



**HAL**  
open science

# Multiphysical modelling and simulation of the ignition transient of complete solid rocket motors

Laurent Francois

► **To cite this version:**

Laurent Francois. Multiphysical modelling and simulation of the ignition transient of complete solid rocket motors. Numerical Analysis [cs.NA]. Institut Polytechnique de Paris, 2022. English. NNT : 2022IPPAX004 . tel-03670668

**HAL Id: tel-03670668**

**<https://theses.hal.science/tel-03670668v1>**

Submitted on 17 May 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT  
POLYTECHNIQUE  
DE PARIS

NNT : 2022IPPAX004

Thèse de doctorat



# Multiphysics modelling and simulation of the ignition transient in solid rocket motors

Thèse de doctorat de l'Institut Polytechnique de Paris  
préparée à l'École Polytechnique

École doctorale n°574  
École Doctorale de Mathématiques Hadamard (EDMH)  
Spécialité de doctorat: Mathématiques aux interfaces

Thèse présentée et soutenue à Palaiseau, le 04/02/2022, par

**Laurent FRANÇOIS**

Composition du Jury :

Vincent GIOVANGIGLI Directeur de recherche, CNRS, CMAP, École Polytechnique	Président du jury
Yves D'ANGELO Professeur, LJAD, Université Côte d'Azur	Rapporteur
Gilles VILMART Maître d'enseignement et de recherche, Section de Mathématiques, Université de Genève	Rapporteur
Stany GALLIER Ingénieur de recherche, ArianeGroup	Examineur
Anne KVAERNO Professeure, Department of Mathematical Sciences, Norwegian University of Science and Technology	Examinatrice
Ronan VICQUELIN Professeur, EM2C, CentraleSupélec	Examineur
Marc MASSOT Professeur, CMAP, École Polytechnique	Directeur de thèse
Joël DUPAYS Ingénieur de recherche, DMPE, ONERA	Encadrant
Paul CLAVIN Professeur émérite, Université de Marseille	Invité
Éric FAUCHER Ingénieur, DGA	Invité





# Remerciements

Une thèse, ça ne se fait pas tout seul ! J'ai eu la chance d'être rudement bien accompagné durant ces 3 années, tant au travail qu'en dehors.

En premier lieu, je remercie chaudement mon directeur de thèse, Marc, pour son encadrement assidu, son foisonnement d'idées, son entrain constant, son humour et les chocolats ! Ça a été un vrai plaisir de travailler avec toi !

À l'ONERA, je voudrais tout d'abord remercier tout aussi chaudement mon encadrant, Joël, toujours accessible, prêt à partager son expérience, ses idées et quelques blagues ! On aura bien rigolé (et bien avancé) dans ton bureau ! Je suis très heureux de bientôt t'avoir comme  $n + 1$  !

D'autres personnes au DMPE ont aussi été essentielles pour ma thèse. Merci Dmitry pour ton expertise, ta rigueur, ta productivité ! Merci à Lionel et Philippe, à jamais condamnés à parcourir les sources de CEDRE. Merci à Robin et Stéphane, de bons échanges et de belles données expérimentales !

Merci aux rapporteurs d'avoir accepté d'éplucher en détail mon travail, et merci aux membres du jury d'avoir accepté de juger mes travaux, avec bienveillance et intérêt !

Qu'aurait été ma thèse sans ce magnifique bureau B01.22 qui, par sa grande baie vitrée et son espace, attire tant de convoitises... Mais surtout, qu'aurait été ce bureau sans mes chers cobureaux : Guillaume S., binôme retrouvé de Supaero, Guillaume P., Aymeric, lauréat des JDD 2022, et les nouveaux arrivants, Margot et Younes. Que de discussions physiques, métaphysiques, et d'entraide ! Je remercie aussi les autres (post-)doctorants des unités MPF et PRA : Albert, Erwin (j'ai pas rendu la chaise !), Florian, Jean-Christophe, Jean-Michel, Jean-Pierre, Maialen, Magdeleine, Mathieu, Pier-Henri, Pierre, Thomas, Valentin, et Virgile. Les pauses café et les croissantages, cela aussi fait pleinement partie de l'expérience de thèse. Merci à notre singulier "stagiaire" permanent, Julien !

À l'X aussi j'ai fait de belles rencontres. Je souhaite tout particulièrement remercier la communauté chrétienne de Polytechnique (CCX) pour son foisonnement d'activités, discussions, repas. Ce temps passé à la CC a été essentiel pour moi. J'aimerais remercier Gonzague, Claire et tous les étudiants que j'y ai rencontrés. Indirectement, la CCX m'a aussi permis de faire la connaissance d'autres personnes que je suis très honoré de pouvoir appeler amis : Bertrand, Brune, Emmanuelle, Marie-Liesse, Maxime, Mikaël, Pomme, Véronique. De belles bastons, de sincères discussions, et bien plus encore ! Merci à vous du fond du cœur.

Mais, par-dessus tout, je veux remercier ma famille, et tout particulièrement mes parents. Ils ont toujours été là pour m'aider dans tous les domaines, bienveillants, attentionnés, aimants. Ils m'ont aussi accueilli à Danjoutin durant les confinements, où j'ai été bien mieux que tout seul dans mon studio !

Enfin, j'aimerais te remercier toi, cher lecteur, qui va lire entièrement (si si!) cette thèse. Je te souhaite bon courage !



# Contents

<b>Remerciements</b>	<b>1</b>
<b>Introduction générale</b>	<b>15</b>
<b>General introduction</b>	<b>27</b>
<b>1 Background on solid rocket motors</b>	<b>37</b>
1.1 Short history and current overview of solid propulsion . . . . .	37
1.1.1 Origins . . . . .	37
1.1.2 Military applications . . . . .	38
1.1.3 Space flight . . . . .	39
1.2 Rocket ballistics . . . . .	40
1.2.1 Structure of a solid rocket motor . . . . .	40
1.2.2 Propulsion principle . . . . .	41
1.2.3 Motor performance . . . . .	42
1.3 Families of solid propellants . . . . .	48
1.3.1 Homogeneous propellants . . . . .	48
1.3.2 Composite propellants . . . . .	49
1.4 Main topics related to solid propulsion . . . . .	50
1.4.1 Unsteady dynamics and instabilities . . . . .	50
1.4.2 Fluid-structure interactions . . . . .	53
1.4.3 Multiphase distributed combustion of aluminium particles . . . . .	53
1.4.4 Numerically-driven propellant formulation . . . . .	55
1.4.5 Numerical studies of heterogeneous propellants . . . . .	56
1.4.6 Transient phenomena . . . . .	57
1.4.7 Electromagnetic signature . . . . .	58
1.4.8 Overview . . . . .	58
1.5 Ignition of SRMs . . . . .	59
1.5.1 Phenomenology of the ignition transient of an SRM . . . . .	59
1.5.2 Solid propellant ignition . . . . .	61
1.5.3 Simulating the ignition transient of an SRM . . . . .	63
1.5.4 Position of the present work . . . . .	68
<b>I One-dimensional modelling and simulation</b>	<b>71</b>
<b>2 One-dimensional solid propellant combustion modelling</b>	<b>75</b>

2.1	General modelling . . . . .	78
2.1.1	Derivation of the model . . . . .	78
2.1.2	The complete model . . . . .	81
2.2	1D modelling of heterogeneous propellants . . . . .	84
2.2.1	Specific heat and density . . . . .	84
2.2.2	Effective thermal conductivity . . . . .	85
2.2.3	Other approaches . . . . .	86
2.2.4	Example result for a typical AP-HTPB blend . . . . .	87
2.3	Pyrolysis laws . . . . .	87
2.3.1	In-depth degradation kinetics . . . . .	88
2.3.2	Asymptotic expansion for the distributed reaction . . . . .	89
2.3.3	Surface reaction . . . . .	89
2.3.4	Shortcomings . . . . .	90
2.4	The ZN approach . . . . .	91
<b>3</b>	<b>Semi-analytical solution in steady-state</b>	<b>95</b>
3.1	Simplification of the generic model . . . . .	98
3.1.1	Physical assumptions . . . . .	98
3.1.2	Mathematical assumptions . . . . .	99
3.1.3	Simplified model . . . . .	100
3.1.4	Conservation properties . . . . .	101
3.1.5	Dimensionless equations . . . . .	103
3.2	Existence and uniqueness of a travelling wave solution profile and velocity . . . . .	105
3.2.1	Monotonicity of the temperature profile . . . . .	105
3.2.2	Existence of a solution . . . . .	106
3.2.3	Uniqueness of the solution . . . . .	108
3.2.4	Heteroclinic orbit and critical points . . . . .	111
3.2.5	Physical interpretation and discussion . . . . .	112
3.3	Numerical method and verification against a CFD code . . . . .	113
3.3.1	Shooting method . . . . .	113
3.3.2	Reference CFD code . . . . .	114
3.3.3	Numerical verification and parametric studies . . . . .	115
3.4	Nonlinear behaviour of $Q_p$ . . . . .	120
<b>4</b>	<b>Semi-discretisation in space and mathematical nature of the obtained system</b>	<b>125</b>
4.1	State of the art for the one-dimensional low-Mach combustion . . . . .	126
4.2	Semi-discretisation in space of the unsteady model . . . . .	127
4.2.1	Gas phase . . . . .	127
4.2.2	Solid Phase . . . . .	129
4.2.3	Surface . . . . .	130
4.3	Differential-algebraic nature of the semi-discretised system . . . . .	130
4.3.1	Identification of the constraints . . . . .	130
4.3.2	Analogy with a singular perturbation problem . . . . .	131
4.3.3	Some background on DAEs . . . . .	132
4.3.4	Index of the semi-discrete system . . . . .	133

<b>5</b>	<b>Numerical strategy</b>	<b>137</b>
5.1	Requirements for the time integration method . . . . .	137
5.1.1	Runge-Kutta methods applied to ODEs . . . . .	137
5.1.2	Convergence of Runge-Kutta methods for index-1 DAEs . . . . .	141
5.1.3	Optimising the computational cost . . . . .	144
5.1.4	Newton algorithm . . . . .	145
5.1.5	Time adaptation . . . . .	145
5.1.6	Final choice of the method . . . . .	148
5.2	Handling the continuity equation . . . . .	149
<b>6</b>	<b>Numerical experiments</b>	<b>155</b>
6.1	Verification . . . . .	155
6.1.1	Verification of the temporal order of convergence . . . . .	156
6.2	Simulation of a laser-induced ignition transient . . . . .	157
6.2.1	Setup . . . . .	157
6.2.2	Results . . . . .	158
6.3	Investigation of limit cycles . . . . .	160
6.3.1	Generating configurations with various degrees of instability . . . . .	160
6.3.2	Methodology . . . . .	163
6.3.3	Analysis of schemes efficiency for a constant time step . . . . .	165
6.3.4	Numerical experiment with time adaptation . . . . .	168
6.4	Application to unsteady combustion with detailed chemistry . . . . .	172
6.4.1	Order of convergence with fixed time steps . . . . .	172
6.4.2	Computational performance with time step adaptation . . . . .	173
<b>II</b>	<b>Coupling</b>	<b>177</b>
<b>7</b>	<b>Coupling the 1D model with a CFD code</b>	<b>181</b>
7.1	Physical models and interactions . . . . .	183
7.1.1	Detailed model . . . . .	183
7.1.2	One-dimensional propellant flame modelling . . . . .	190
7.2	Coupled framework for SRM ignition simulations . . . . .	191
7.2.1	Description of the CFD solver . . . . .	191
7.2.2	Coupling methodology . . . . .	192
7.2.3	Coupling in time . . . . .	193
7.3	Modelling issues . . . . .	194
7.3.1	Conservativity, time lag and flame quasi-steadiness . . . . .	194
7.3.2	Transition of the wall heat flux during ignition . . . . .	196
7.4	One-dimensional verification . . . . .	198
<b>8</b>	<b>Towards a higher-order adaptive coupling</b>	<b>203</b>
8.1	Co-simulation . . . . .	207
8.1.1	Integration algorithm . . . . .	209
8.1.2	Local truncation error . . . . .	210
8.1.3	Time step adaptation . . . . .	211
8.2	A toy model . . . . .	212

8.2.1	Model . . . . .	212
8.2.2	Semi-discretisation in space . . . . .	213
8.2.3	Imposing the surface connection condition . . . . .	214
8.2.4	Matrix form of the equations . . . . .	215
8.3	Co-simulation framework . . . . .	216
8.3.1	Transmission condition . . . . .	216
8.3.2	Coupling procedure . . . . .	218
8.4	Assessment of the convergence . . . . .	218
8.4.1	Dynamics of the toy model . . . . .	218
8.4.2	Co-simulation results . . . . .	220
8.5	Time step and order adaptation . . . . .	222
8.5.1	Dynamics of the 0D chamber ignition . . . . .	222
8.5.2	Effect of the prediction order . . . . .	223
8.5.3	Order adaptation . . . . .	224

### **III Applications 227**

#### **9 Model fitting 231**

9.1	Experimental data available at ONERA . . . . .	232
9.1.1	Thermal and radiative measurements . . . . .	232
9.1.2	Closed bomb . . . . .	233
9.1.3	Ignition dynamics . . . . .	234
9.1.4	Other characterisations . . . . .	236
9.2	From experimental data to model parameters . . . . .	238
9.2.1	Gas phase parameters obtained from equilibrium considerations . . . . .	238
9.2.2	Further parameter relations and bounds . . . . .	239
9.2.3	Free parameters . . . . .	240
9.3	Optimisation framework . . . . .	240
9.3.1	Definition of the optimisation problem . . . . .	240
9.3.2	Optimisation algorithm . . . . .	242
9.4	Results . . . . .	243
9.4.1	Optimised parameters . . . . .	243
9.4.2	Sensitivity of the propellant characteristics . . . . .	244

#### **10 Code developments 247**

10.1	Standalone solvers . . . . .	247
10.1.1	Semi-analytical tool . . . . .	247
10.1.2	One-dimensional unsteady propellant solvers . . . . .	248
10.2	Python frameworks . . . . .	253
10.2.1	Automatic parametric studies . . . . .	253
10.2.2	Optimisation of model parameters . . . . .	254
10.3	Developments in CEDRE . . . . .	255
10.3.1	1D flame approach . . . . .	255
10.3.2	Detailed approach . . . . .	256
10.4	Demonstrators of co-simulation . . . . .	256

<b>11 Numerical assessment of the coupled approaches on a newly designed test case for two-dimensional ignition</b>	<b>259</b>
11.1 Definition of the test case . . . . .	260
11.1.1 Geometry and boundary conditions . . . . .	263
11.1.2 Initial conditions . . . . .	264
11.1.3 Spatial discretisation . . . . .	264
11.1.4 Numerical setup . . . . .	266
11.2 Reference result obtained with the detailed approach . . . . .	266
11.2.1 Dynamics of the combustion chamber . . . . .	267
11.2.2 Ignition dynamics . . . . .	270
11.3 Comparison of both modelling approaches . . . . .	270
11.3.1 Integration with a fine time step . . . . .	270
11.3.2 Influence of the temporal resolution . . . . .	274
11.3.3 Investigation of the instability for the 1D flame approach . . . . .	279
 <b>Conclusions and perspectives</b>	 <b>285</b>



# List of Figures

1	Vue 3D de la tuyère et de la forme initiale du grain de propergol dans le moteur P80 du lanceur Vega (©ArianeGroup) . . . . .	16
2	Principales phases du transitoire d’allumage d’un MPS axisymétrique. De haut en bas : démarrage de l’allumeur, premier allumage de la charge principale, propagation de la flamme, déformation mécanique et rupture de l’opercule de la tuyère. . . . .	17
3	Schéma des différents phénomènes intervenant dans la combustion des propergols solides [38] . . . . .	19
4	Schéma du modèle d’allumage quasi-unidimensionnel. La direction de l’écoulement de la chambre est l’axe longitudinal en bleu. Les segments verticaux en pointillés bleus montrent les limites des cellules du maillage discrétisant la chambre. Les modèles unidimensionnels de la combustion du propergol sont représentés pour les 3 premières cellules, et les courbes rouges représentent le profil de température local dans la phase solide et la flamme. . . . .	20
5	3D view of the nozzle and initial propellant grain shape in the P80 motor of the Vega launcher (©ArianeGroup) . . . . .	27
6	Main phases of the ignition transient of an axisymmetric SRM. From top to bottom: igniter start-up and impingement, first ignition of the main load, flame spreading, mechanical deformation and rupture of the nozzle cap. . . . .	29
7	Phenomena involved in a solid propellant combustion wave [38] . . . . .	30
8	Sketch of the one-dimensional ignition SRM model: the axis for the quasi-1D chamber flow is in blue, and the blue dashed vertical segments show the limits of the CFD cells discretising the chamber. The 1D propellant models are shown for the first 3 cells, and the red curves indicate the local propellant temperature profile. . .	32
1.1	Notorious examples of solid rockets used in the military . . . . .	39
1.2	Examples of civil launchers using SRMs . . . . .	40
1.3	Cutaway of a typical solid rocket motor [6] . . . . .	41
1.4	Conservation of momentum for an idealised rocket . . . . .	41
1.5	Measured burning rates for six monopropellants [85] . . . . .	43
1.6	Regression of a star-shaped solid propellant grain (own work [86]) . . . . .	44
1.7	Simplified schematic of the internal flow of an SRM . . . . .	45
1.8	Influence of the initial geometric configuration on the thrust profile during the firing [87] . . . . .	46
1.9	Cutaway of a typical second stage of an early French ballistic missile [77] . . . . .	47
1.10	Structure of the flame for an AP-HTPB composite propellant [6] . . . . .	50
1.11	Left: Pressure signal (black) and fluctuating part (red) [89]. Right: decomposition of the fluctuations on the first 3 longitudinal modes. . . . .	51

1.12	Typical response function $R_{mp}$ for a solid propellant (left: real part, right: imaginary part) [91] . . . . .	52
1.13	Ombrosopy visualisation of aluminium droplets burning above the propellant surface (©ONERA, courtesy of Robin Devilliers) . . . . .	54
1.14	Simulated temperature field around a burning aluminium particle of diameter 120 $\mu\text{m}$ with an alumina lobe (in grey) [112] . . . . .	55
1.15	Pack of $10^5$ AP particles [124] . . . . .	56
1.16	Typical pressure history during ignition for large or low port-to-throat area ratio motors[46] . . . . .	59
1.17	Snapshot of the surface temperature distribution in the Space Shuttle SRM during ignition as simulated with ROCSTAR [154] . . . . .	67
2.1	One-dimensional model of solid propellant combustion . . . . .	79
2.2	Two simple homogenisation approaches for the determination of the effective thermal conductivity . . . . .	86
3.1	Schematic phase portrait in both phases . . . . .	107
3.2	Evolution of the phase portrait with $m$ . Each curve corresponds to the solution curve for one value of $m$ . . . . .	107
3.3	Plot of $\theta$ and $Y$ for the semi-analytical solution (dots) and the numerical simulation (dashed lines) . . . . .	116
3.4	Convergence of the CFD solution towards the semi-analytical solution with an adapted mesh . . . . .	116
3.5	Temperature profiles, CFD results (full lines) compared to semi-analytical results (circles) . . . . .	118
3.6	Normalised phase portraits, CFD results (full lines) compared to semi-analytical results (circles) . . . . .	118
3.7	Regression speed as a function of activation temperature, CFD results compared to semi-analytical and analytical results . . . . .	119
3.8	Evolution of the regression speed with $c_p/c_c$ obtained with the semi-analytical and CFD methods . . . . .	119
3.9	Effects of the ratio $c_p/c_c$ on $\xi$ and on the solution regression speed . . . . .	119
3.10	Temperature profiles, CFD results (full lines) compared to semi-analytical results (dots) for different values of the Lewis number . . . . .	120
3.11	Relative error on the prediction of $r$ by the semi-analytical method compared to the CFD method, as a function of the Lewis number . . . . .	120
3.12	Evolution of $\xi$ in the case of heat loss by radiation at the surface, for different values of $\epsilon$ . . . . .	122
3.13	Solutions obtained with $\epsilon = 50$ . . . . .	122
4.1	Localisation of the discretised variables in the finite volume mesh. The thick vertical line represents the surface. . . . .	128
4.2	Sparsity pattern of the Jacobian of $g$ , the vector of constraints . . . . .	134
6.1	Bode diagram of the response function $R_{mp}$ to pressure fluctuations . . . . .	156
6.2	Surface temperature histories obtained with IE when gradually lowering the time step	157
6.3	Global error $\epsilon_{\rho u}$ on the mass flow rate field . . . . .	157

6.4	Ignition transient computed with various methods. . . . .	159
6.5	Work-precision diagram for the determination of $t_{ign}$ : (a) coarse mesh (400 cells), (b) fine mesh (2049 cells). . . . .	159
6.6	Generation of unstable configurations: (a) Segment travelled in the (r, k) stability diagram, (b) Unsteady simulations of some configurations (ESDIRK-54A with $rtol = 10^{-6}$ ) . . . . .	163
6.7	Main features of the studied configuration . . . . .	164
6.8	Envelopes of the surface temperature histories computed for different time steps and integration methods . . . . .	166
6.9	Fitted amplification factor . . . . .	167
6.10	Convergence of the limit cycle properties . . . . .	167
6.11	Work-precision diagrams for fixed time step simulations . . . . .	168
6.12	Convergence and computational cost for the amplification factor . . . . .	169
6.13	Limit cycle with adaptive time stepping: (a) convergence of the fundamental amplitude, (b) work-precision diagram . . . . .	170
6.14	Comparison of the computational cost for a given level of relative error . . . . .	171
6.15	Accuracy of the integration with fixed time steps: (a) Convergence of $\epsilon_{T_s}$ , (b) Work-precision diagram for $\epsilon_{T_s}$ . . . . .	173
6.16	Integration with adaptive time stepping: (a) time step evolution for ESDIRK-43B, (b) work-precision diagram for the time integral error $\epsilon_{T_s}$ . . . . .	174
7.1	Artist's view of the coupling process . . . . .	181
7.2	Sketch of the connection conditions between the 1D model and the CFD mesh . . . . .	189
7.3	Coupling algorithm between VULC1D and CHARME . . . . .	193
7.4	Illustration of the spatial coupling between the 1D code (upper part) and the CFD code (lower part), showing the overlap of the 1D flame and the CFD domain . . . . .	195
7.5	Typical surface evolution for a single propellant point during a simulated chamber ignition . . . . .	199
7.6	Comparison of the surface temperature evolutions for the laser-induced ignition . . . . .	200
8.1	Mesh used for the toy model . . . . .	213
8.2	Dynamics of the toy model . . . . .	220
8.3	Convergence of the global error for different prediction orders $p$ . Grey dashed curves corresponds to the theoretical orders. . . . .	221
8.4	Coupled dynamics of the solid propellant and the combustion chamber. . . . .	223
8.5	Visualisation of the predictive polynomials for various orders, around the time of ignition (laser cut-off). The black dots represent the surface temperature computed by VULC1D at each coupling step. . . . .	223
8.6	Evolution of the time step for various prediction orders. . . . .	224
8.7	Evolution of the prediction order . . . . .	226
9.1	Measurement of the propellant surface emissivity by comparison with a reference paint	233
9.2	Pressure and regression rates measured within a closed bomb (coloured lines), fitted Vieille law (black line) . . . . .	234
9.3	Geometry and experimental setup of the CHAMADE test bench [25] . . . . .	234
9.4	Different phases of laser-induced ignition, as recorded by an infrared camera (courtesy of Robin Devillers, ONERA) . . . . .	235

9.5	Surface temperature history deduced from infrared images (courtesy of Robin Devillers, ONERA) . . . . .	236
9.6	Example ignition law obtained experimentally (courtesy of Robin Devillers, ONERA)	236
9.7	Schematic view of the modulated exhaust test bench [292] . . . . .	237
9.8	Surface temperature evolution for 3 different heat flux levels and expected ignition times (vertical dashed lines) based on the experimental ignition law . . . . .	244
11.1	Successive evolutions of the test case. Large arrows indicate subsonic outflow boundaries. . . . .	261
11.2	Geometry and principle of the test case (temperature field at $t = 2$ ms) . . . . .	263
11.3	Field of equivalent cell diameter for the fine mesh . . . . .	265
11.4	Details of the fine mesh near the propellant surface and ignition front . . . . .	265
11.5	Evolution of the temperature field in the reference simulation . . . . .	268
11.6	Zoom on the flow field around the ignited zones at $t = 5$ ms with the detailed approach on the fine mesh . . . . .	269
11.7	Evolution of the propellant surface for the reference simulation . . . . .	270
11.8	Evolution of the temperature field with $\Delta t = 2 \times 10^{-8}$ s. From left to right: detailed coupling, 1D flame coupling ( $\delta = 4 \mu\text{m}$ ), 1D flame coupling ( $\delta = 20 \mu\text{m}$ ). From top to bottom: $t = 1, 2, 3, 4, 5$ ms. . . . .	272
11.9	Velocity field near $x = 0.01$ m at $t = 5$ ms . . . . .	273
11.10	Detailed approach ( $\delta = 1 \mu\text{m}$ ) with $\Delta t = 2 \times 10^{-8}$ s . . . . .	275
11.11	1D flame coupled approach ( $\delta = 4 \mu\text{m}$ ) with $\Delta t = 2 \times 10^{-8}$ s . . . . .	275
11.12	1D flame coupled approach ( $\delta = 20 \mu\text{m}$ ) with $\Delta t = 2 \times 10^{-8}$ s . . . . .	275
11.13	Detailed approach ( $\delta = 1 \mu\text{m}$ ) with $\Delta t \leq 2 \times 10^{-6}$ s . . . . .	277
11.14	1D flame coupled approach ( $\delta = 4 \mu\text{m}$ ) with $\Delta t \leq 2 \times 10^{-6}$ s . . . . .	277
11.15	1D flame coupled approach ( $\delta = 20 \mu\text{m}$ ) with $\Delta t \leq 2 \times 10^{-6}$ s . . . . .	277
11.16	Heat flux entering the solid phase at $x = 7.8 \times 10^{-3}$ m . . . . .	278
11.17	Relative difference of the solid phase heat flux compared to the reference simulation at $x = 7.8 \times 10^{-3}$ m . . . . .	278
11.18	Evolution of the CFD time step . . . . .	279
11.19	Temporal evolution of the outermost ignited positions . . . . .	280

# List of Tables

2.1	Input data . . . . .	87
2.2	Homogenised for properties for an 86 % AP - 14 % HTPB mixture . . . . .	87
5.1	Butcher tableau of the ESDIRK-54A method [68] . . . . .	154
6.1	Selected Runge-Kutta methods. The numbers in brackets correspond to the number of stages actually solved. . . . .	158
8.1	Number of time steps taken for each prediction order . . . . .	225
9.1	Relative sensitivities of various characteristics with respect to the optimised model parameters at convergence . . . . .	244
11.1	Main characteristics of the 3 meshes considered . . . . .	266



# Introduction générale

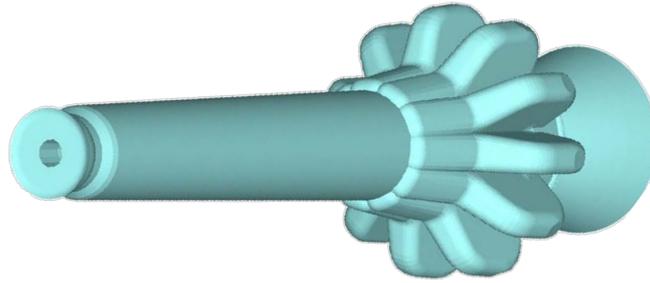
L'avènement de la propulsion spatiale est l'un des grands progrès technologiques du XX<sup>ème</sup> siècle. Les fusées sont aujourd'hui utilisées couramment pour de multiples applications, allant des lanceurs spatiaux pour le déploiement de satellites ou les vols habités, aux systèmes de propulsion pour différents types de missiles. L'élément le plus important d'une fusée est son moteur, qui génère la poussée nécessaire à l'accélération du véhicule et de sa charge utile. Il existe plusieurs types de moteurs-fusées, les principaux étant les moteurs à ergols liquides et les moteurs à propergol solide (MPS) [1].

Les MPS utilisent un propergol solide, qui est fabriqué généralement sous la forme d'un cylindre, au centre duquel une extrusion initiale constitue la chambre de combustion. Après un processus d'allumage, la surface du propergol solide se décompose et se gazéifie. Les produits gazeux qui en résultent brûlent sans aucune injection externe de combustible ou de réactif. Les MPS présentent plusieurs avantages par rapport aux autres systèmes :

- de très grands moteurs peuvent être fabriqués, ce qui permet de générer des niveaux de poussée très élevés, par exemple jusqu'à 1200 tonnes pour chaque booster de la navette spatiale américaine ;
- ils sont beaucoup plus simples en termes de structure, car ils comportent notamment très peu de pièces mobiles ;
- ils peuvent être stockés pendant de longues périodes sans maintenance spécifique ;
- ils peuvent être allumés rapidement à la demande, sans le complexe processus de refroidissement nécessaire pour les moteurs à ergols liquides.

Ainsi, ils sont utilisés sur un certain nombre de lanceurs lourds, par exemple Ariane 5, la future Ariane 6 et le SLS (Space Launch System), ainsi que sur des fusées plus petites à faible coût, par exemple le lanceur italien Vega ou de petites fusées-sondes utilisées pour des expériences suborbitales. Les avantages précédents les rendent également particulièrement appropriés pour des usages militaires, tant pour les petits systèmes d'armes, comme les lances-roquettes, que pour les missiles balistiques.

Depuis les années 1950, la recherche d'une amélioration des performances et de la fiabilité a motivé de nombreuses recherches. Les expériences sur des moteurs à échelle 1 sont cependant coûteuses et longues, et elles ne peuvent pas être utilisées pour effectuer des études paramétriques, qui permettraient de connaître la sensibilité d'un moteur à différents paramètres de conception par exemple. La modélisation théorique simplifiée des MPS a permis de mieux comprendre divers sujets, comme les instabilités de pression [2, 3], l'allumage [4], l'évolution de la géométrie interne de la chambre lors d'un tir [5]. Cependant, les moteurs réels sont souvent trop complexes en raison du nombre et de la nature des phénomènes physiques impliqués, ainsi que de la géométrie interne de la charge propulsive (ou grain), comme le montre par exemple la figure 1. Par conséquent, leur dynamique ne peut être précisément reproduite ou prédite à l'aide de modèles simples. C'est pourquoi la simulation numérique de modèles plus complets, dont l'analyse théorique est hors de



**Figure 1** Vue 3D de la tuyère et de la forme initiale du grain de propergol dans le moteur P80 du lanceur Vega (©ArianeGroup)

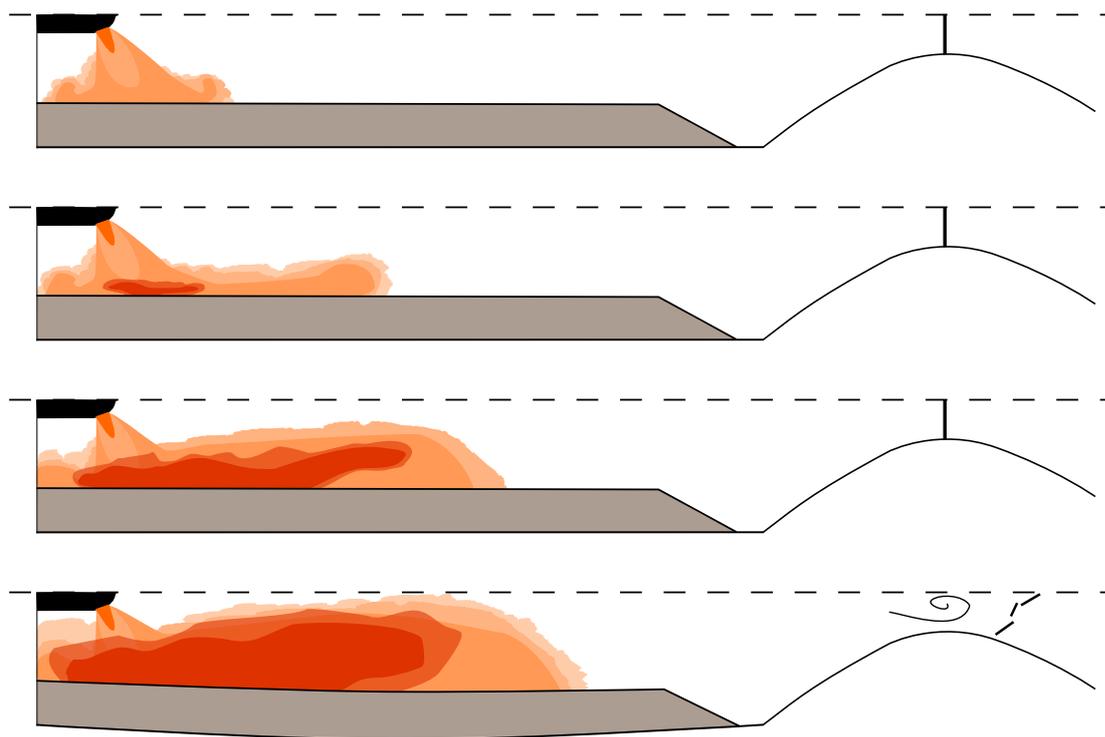
portée, a suscité beaucoup d'attention depuis les années 1970, lorsque la puissance de calcul et les méthodes numériques sont devenues adéquates.

Aujourd'hui, il existe un grand nombre d'outils de simulation des MPS pour une variété de sujets. Par exemple, l'évolution de la poussée pendant un tir complet peut être prédite avec précision [6]. En particulier, des outils permettent de calculer l'évolution de la géométrie interne, au fur et à mesure que le propergol solide s'épuise [7, 8]. Les instabilités de pression dans la chambre, liées aux interactions entre l'hydrodynamique, l'acoustique et la combustion du propergol solide ont été étudiées principalement par simulation [9–11]. Les effets du caractère multiphasique de l'écoulement à l'intérieur de la chambre de combustion ont donné lieu à un grand nombre de travaux numériques [11–15], et de nombreuses recherches sont encore consacrées à ce sujet. La complexité multiphysique de l'estimation de la signature électromagnétique des panaches de MPS a également été étudiée [16]. Pour améliorer les performances de différentes formulations chimiques de propergol, des mécanismes de réaction détaillés ont été développés [17, 18], et des simulations de la microstructure des propergols solides composites (une classe importante de propergols solides, où le matériau est hétérogène) ont également été présentées [19–21].

Grâce aux travaux de recherche et de développement approfondis menés depuis les années 1950, les MPS sont devenus le mode de propulsion à forte poussée le plus fiable, avec pratiquement aucune défaillance de moteur en vol aux États-Unis et en Europe au cours des vingt dernières années [22]. En fait, la seule défaillance due à un moteur de fusée à propergol solide au cours de cette période est survenue lors de la mission Vega VV-15, où le sous-dimensionnement d'une protection thermique dans le MPS du deuxième étage a entraîné l'explosion du lanceur [23]. Malgré le taux de fiabilité important, un grand nombre des phénomènes physiques présents dans les MPS sont encore mal compris et prédits.

Un sujet d'intérêt particulier est celui de l'allumage. Il s'agit du processus d'initiation de la combustion de la charge propulsive. Un propergol solide brûle généralement à des températures de surface de l'ordre de 700 à 1000 K. Avant un tir, un MPS est généralement au repos à température ambiante. Ainsi, la surface du propergol doit d'abord être chauffée. Cela se fait habituellement en utilisant un moteur plus petit (allumeur pyrogène), pour générer un grand volume de gaz chauds ( $> 2000$  K), qui sont expulsés à grande vitesse vers la charge principale. Il s'ensuit un chauffage non uniforme de cette dernière, qui amène d'abord à l'allumage de quelques points sur la surface de la charge principale. La flamme de propergol qui apparaît se propage ensuite au reste de la surface.

La phase d'allumage est critique, car le MPS subit de forts gradients spatiaux et temporels de pression, température et vitesse à l'intérieur de la chambre de combustion. En particulier, l'évolution de la pression pendant cette phase est un facteur dimensionnant de la structure mécanique du MPS.



**Figure 2** Principales phases du transitoire d’allumage d’un MPS axisymétrique. De haut en bas : démarrage de l’allumeur, premier allumage de la charge principale, propagation de la flamme, déformation mécanique et rupture de l’opercule de la tuyère.

Pour garantir un fonctionnement sûr, cette pression ne doit pas dépasser les limites de conception, afin de ne pas provoquer de défaillance structurelle ou d’endommager la charge propulsive, ce qui aurait un impact sur le niveau de poussée du MPS après l’allumage.

Cela nécessite un dimensionnement adéquat du système d’allumage, de sa charge et de l’énergie qu’il délivre. Le grand volume de gaz chauds générés par l’allumeur est dirigé vers la charge principale par de petites buses, dont l’orientation doit être choisie de manière appropriée. Le système d’allumage doit garantir un allumage complet et rapide, afin d’éviter les ratés qui pourraient entraîner une combustion lente et ininterrompue du propergol. L’augmentation de la pression pendant l’allumage doit également être suffisamment lente pour que les déformations structurales n’entraînent pas la formation de fissures dans le propergol, la rupture de certaines parties de la charge, ou une mauvaise adhérence entre le chargement et l’enveloppe du moteur, ce qui pourrait entraîner l’accumulation de gaz chauds susceptibles de provoquer un allumage indésirable et la rupture de l’enveloppe à la suite d’une surpression. En outre, la chambre de combustion des MPS est généralement scellée hermétiquement avant l’allumage. L’opercule assurant l’étanchéité initiale doit être conçu pour se rompre à une pression suffisamment élevée, de sorte que les gaz d’allumage restent plus longtemps dans la chambre de combustion, améliorant ainsi le transfert d’énergie vers la charge principale. La rupture doit être propre et ne pas conduire à une obstruction partielle de la tuyère, et les oscillations de pression associées doivent être bien contenues. Enfin, l’allumage du grain principal doit être symétrique, afin d’éviter une consommation non uniforme du propergol, qui pourrait conduire à une poussée asymétrique en fin de combustion.

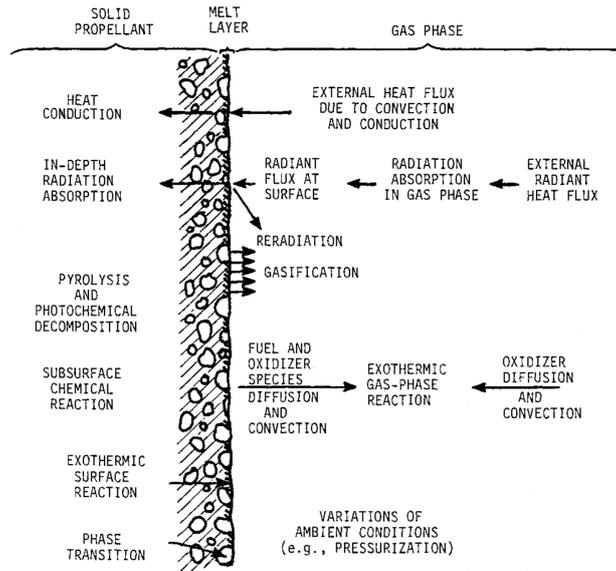
Les différentes phases d’un transitoire d’allumage sont représentées dans la figure 2. Le

transitoire d'allumage implique une variété de phénomènes, parmi lesquels : l'échauffement du propergol par contact avec le flux de l'allumeur, la pyrolyse du propergol à sa surface (production d'espèces gazeuses), la combustion rapide des produits de pyrolyse au voisinage immédiat de la surface, les phénomènes acoustiques et hydrodynamiques à l'intérieur de la chambre de combustion, l'écoulement multiphasique (présence de particules liquides et solides), la large gamme de vitesses d'écoulement (1-1000 m/s), le rayonnement thermique des particules, de la flamme et des différentes surfaces, la déformation de la charge propulsive, la rupture de l'opercule... Tous ces phénomènes contribuent aux caractères hautement multiphysique et transitoire de l'allumage des MPS et de leur balistique interne. De plus, l'allumage est fortement dépendant de la géométrie interne du grain de propergol, une partie de la surface du propergol pouvant être très peu exposée au flux des gaz enflammés. Ces aspects rendent complexes la compréhension et la prévision du processus d'allumage. Il est essentiel de pouvoir caractériser correctement l'allumage et, pour améliorer le processus de conception d'un MPS, de pouvoir le prédire avec précision à l'aide d'outils de simulation numérique.

Un ingrédient important de la prédiction du transitoire d'allumage d'un MPS est la prédiction de l'allumage du propergol solide lui-même. Expérimentalement, ceci peut être étudié en plaçant de petits échantillons de propergol dans un environnement contrôlé, et en les exposant à un flux convectif (impact de gaz chauds) [24] ou à un flux laser [25]. Cela permet de déterminer la loi d'allumage d'un propergol, c'est-à-dire le temps requis pour qu'il s'allume lorsque sa surface est exposée à un flux thermique approximativement constant.

Cependant, le flux thermique à la surface du propergol dans un MPS varie largement dans l'espace et dans le temps pendant le transitoire d'allumage. Par conséquent, un modèle dynamique est nécessaire pour calculer l'allumage du propergol dans ce cas. À cette fin, il est pratique de considérer une représentation unidimensionnelle du propergol solide dans la direction perpendiculaire à la surface, car les divers phénomènes (conduction thermique, transfert de chaleur conjugué avec l'écoulement environnant) se produisent principalement dans cette direction. Les premiers modèles d'allumage [4, 26] utilisaient une représentation simplifiée des différents phénomènes : flux de chaleur constant, omission de la phase gazeuse. Leur simplicité permettait une résolution analytique. Les modèles ultérieurs se sont appuyés sur une résolution numérique de la conduction thermique instationnaire à l'intérieur du solide et sur des modèles de flamme de propergol simplifiés, basés soit sur une limite d'énergie d'action élevée [27], soit sur un modèle plus générique mais plus artificiel [28, 29], où la distribution du dégagement de chaleur en phase gazeuse est prescrite, au lieu d'utiliser un modèle d'écoulement réactif pour résoudre le profil de flamme associé. Pour contourner le problème de la modélisation de la flamme du propergol, l'approche de Zeldovich-Novozhilov (ZN) [30] élimine la partie du modèle en phase gazeuse et la remplace par des relations empiriques déduites d'expériences, mais ces données sont généralement imprécises et insuffisantes pour une simulation correcte de l'allumage sans s'appuyer sur des modèles supplémentaires [31].

Certaines études ont été présentées pour l'allumage d'un propergol solide considéré unidimensionnel, en utilisant une description détaillée des phénomènes de surface du propergol (présence d'une fine couche liquide) et une cinétique complexe en phase gazeuse [18, 32–34]. Les outils de simulation associés sont la plupart du temps des adaptations de codes développés pour l'étude du régime permanent ou instationnaire [18, 35–37]. Tous ces outils résolvent plusieurs systèmes d'équations couplés de manière forte : conduction thermique instationnaire dans le propergol solide, dégradation et décomposition en profondeur du propergol au sein de la phase solide, fusion des produits de décomposition dans la couche liquide superficielle, évaporation de cette couche et combustion des produits évaporés dans la phase gazeuse à faible nombre de Mach.



**Figure 3** Schéma des différents phénomènes intervenant dans la combustion des propergols solides [38]

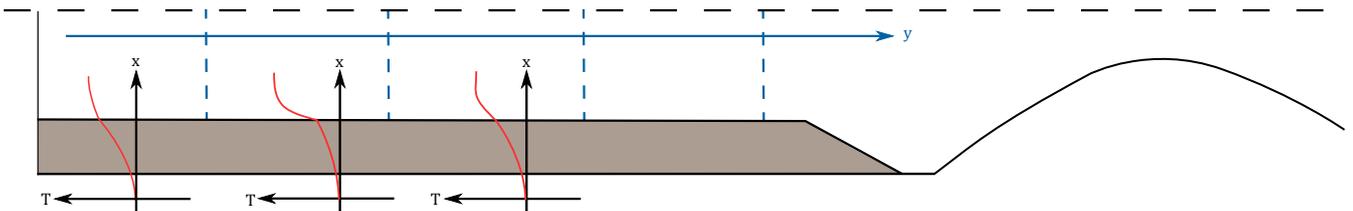
Le grand nombre de processus intervenant dans la combustion d'un propergol est résumé dans la figure 3.

Il faut noter que peu de travaux ont été consacrés à l'étude de l'allumage pour un modèle géométriquement détaillé de propergols solides composites. En raison du coût de calcul, de telles simulations ne sont réalisables qu'avec une cinétique globale simplifiée [21, 39]. Les outils détaillés unidimensionnels abordés précédemment sont d'un grand intérêt pour étudier les effets des processus physico-chimiques fondamentaux. Cependant, ils sont très complexes et leur résolution numérique repose sur l'intégration séparée des phases gazeuse et solide, avec divers algorithmes d'intégration d'ordre inférieur pour chaque sous-problème, couplés itérativement. **Le couplage entre les différents solveurs est du premier ordre en temps**, ce qui est sous-optimal pour la performance de l'outil de simulation complet. De plus, **la stabilité limitée empêche l'utilisation de maillages fortement raffinés en espace** [33], ce qui limite la précision de la discrétisation spatiale. Une partie du problème provient du **manque d'analyse minutieuse de la nature mathématique du système global d'équations discrètes** à résoudre. Ce dernier est de nature différentielle-algébrique d'index 1, pour laquelle le schéma temporel Euler implicite du premier ordre, souvent utilisé dans les communautés de la combustion unidimensionnelle des propergols solides et des flammes homogènes à faible nombre de Mach, fonctionne bien. Cependant, la généralisation à des ordres supérieurs n'a pas encore été correctement abordée. Disposer d'une stratégie numérique d'ordre élevé est important pour améliorer l'efficacité des calculs et mieux capturer la dynamique du système simulé. En effet, l'avantage des méthodes d'ordre supérieur a été clairement identifié dans une variété de travaux pour les simulations haute-fidélité d'autres systèmes dynamiques [40, 41].

Revenons sur le sujet de la simulation du transitoire d'allumage dans les MPS. Un effort de recherche important a été mené dans cette direction depuis les années 1950. Les premières études s'appuyaient sur des modèles simplifiés qui supposaient une représentation en dimension zéro de la chambre de combustion [42–45]. Un certain nombre de paramètres ad hoc devaient être ajustés pour correspondre correctement aux résultats expérimentaux. Cette approche ne pouvait donc pas

être utilisée de manière prédictive, en particulier pour les moteurs de longueur importante, où le champ de pression n'est pas uniforme dans la chambre.

Dans les années 1970, des méthodes numériques ont commencé à apparaître pour la résolution des équations d'Euler instationnaires, permettant de résoudre l'évolution instationnaire de l'écoulement compressible à l'intérieur du moteur, initialement avec des solveurs CFD quasi-unidimensionnels [46], où la coordonnée spatiale unique pour le modèle d'écoulement de la chambre est selon l'axe longitudinal du moteur. Les simulations tenant compte des non-uniformités spatiales le long de la chambre (par exemple la pression) ont permis d'augmenter considérablement la fidélité des résultats numériques par rapport aux expériences [46, 47]. Pour reproduire l'échauffement progressif du propergol, la conduction thermique instationnaire dans le solide est prise en compte. Suite à l'observation que la conduction thermique se produit principalement dans la direction perpendiculaire à la surface du propergol [43], la modélisation de la conduction thermique et de la combustion du propergol a été simplifiée à une approche unidimensionnelle. À chaque cellule du maillage spatial quasi-unidimensionnel de l'écoulement de la chambre, une équation de chaleur unidimensionnelle est utilisée pour modéliser le chauffage en profondeur du propergol dans la direction perpendiculaire à la surface du propergol, c'est-à-dire généralement perpendiculaire à l'axe longitudinal du moteur. Plusieurs instances du modèle d'échauffement sont utilisées sur la longueur du moteur pour tenir compte du transfert de chaleur et de la température en surface du propergol non uniformes spatialement, comme le montre la figure 4.



**Figure 4** Schéma du modèle d'allumage quasi-unidimensionnel. La direction de l'écoulement de la chambre est l'axe longitudinal en bleu. Les segments verticaux en pointillés bleus montrent les limites des cellules du maillage discrétisant la chambre. Les modèles unidimensionnels de la combustion du propergol sont représentés pour les 3 premières cellules, et les courbes rouges représentent le profil de température local dans la phase solide et la flamme.

Ce modèle pourrait en théorie être n'importe lequel des modèles d'allumage unidimensionnels mentionnés précédemment, qui diffèrent principalement par leur représentation de la flamme du propergol. Cependant, afin d'optimiser le coût de calcul et de se baser uniquement sur une paramétrisation simple, les modèles d'allumage choisis en pratique sont beaucoup plus simples. La conduction thermique instationnaire dans le solide est toujours résolue comme dans les modèles précédents, mais la flamme du propergol n'est pas prise en compte. Au lieu de cela, le propergol est supposé inerte, tant que la température de surface est inférieure à une température d'allumage spécifiée par l'utilisateur. Une fois cette valeur atteinte, le modèle unidimensionnel de conduction thermique en phase solide est désactivé, et l'on suppose que le propergol s'allume instantanément et qu'il brûle de manière quasi-stationnaire par rapport à l'écoulement environnant.

Des évolutions de cette modélisation unidimensionnelle de l'allumage des MPS sont encore largement utilisées aujourd'hui [48, 49], en utilisant des descriptions plus raffinées de l'impact du flux de l'allumeur sur la surface du propergol et du transfert de chaleur par rayonnement thermique. Grâce à la simplicité de ces modèles, des études paramétriques autour de configurations connues

peuvent être réalisées à un faible coût de calcul. Cependant, ces outils ne peuvent pas être utilisés pour prédire le comportement d'un nouveau moteur. Un grand nombre d'expériences est nécessaire pour ajuster correctement certains paramètres du modèle, en particulier en ce qui concerne le transfert de chaleur dans les zones où la charge propulsive est fortement tridimensionnelle, comme montré en figure 1, ce qui est le cas dans de nombreux MPS.

Il est donc nécessaire de passer à des modèles 2D ou 3D de l'écoulement dans la chambre de combustion. En raison du coût de calcul important associé à de telles simulations, les premières études ne sont apparues que dans les années 1990 [50–52]. Pour reproduire l'évolution de la pression observée expérimentalement sur divers moteurs, le flux de chaleur radiatif émanant de la flamme du propergol a été signalé comme un facteur important pour une prédiction correcte de la propagation de la flamme [52]. Ainsi, certaines zones, telles que les petits espaces présents entre différents segments de la charge propulsive, s'enflamment plus rapidement, car elles sont autrement peu exposées au flux convectif de l'allumeur. Un important programme de recherche et développement a été mené au CSAR (Center for Simulation of Advanced Rockets) dans les années 2000 et a conduit au développement de ROCSTAR, une suite logicielle capable de simuler l'allumage des MPS et de prendre en compte la déformation dynamique de la charge propulsive [7, 53–55]. Dans tous ces travaux, le modèle d'allumage du propergol reste le même que dans les études antérieures sur l'allumage des MPS. Le propergol solide se comporte initialement comme un matériau inerte et, lorsqu'il atteint une température de surface d'allumage spécifiée par l'utilisateur, il passe instantanément à un comportement de combustion quasi-stationnaire, uniquement affecté par la pression pariétale locale prescrite par le solveur CFD. La flamme du propergol n'est pas modélisée, mais la température de la flamme est calculée *a priori*, en supposant l'équilibre chimique. Des outils similaires ont été développés récemment en Chine [56–59] avec la même modélisation de l'allumage.

Ces modèles souffrent de plusieurs limitations. La première est **l'utilisation d'une température d'allumage constante prédéfinie** pour déterminer si le propergol est localement allumé. Il a été démontré que cette température dépend en réalité grandement de l'historique du flux thermique de la paroi et, plus faiblement, de la pression [27, 60]. Une deuxième limite est que toutes les simulations d'allumage rapportées couplent les solveurs d'écoulement de fluide et de propergol avec une **précision du premier ordre en temps**, ce qui peut avoir un impact sur la précision de la solution et l'efficacité du calcul. Une troisième limite est la **précision restreinte de la représentation de la flamme du propergol**. En effet, du point de vue du solveur CFD, la flamme est modélisée comme un phénomène de surface. En réalité, elle a une épaisseur de l'ordre de 100  $\mu\text{m}$ . Le fait de comprimer cette flamme en un phénomène de surface peut empêcher certaines interactions physiques complexes avec l'écoulement environnant. En effet, même si les résultats obtenus avec cette approche de flamme unidimensionnelle rapportés dans la littérature sont généralement en bon accord avec les données expérimentales, cela ne constitue pas une vérification complète de la cohérence physique de cette simplification. En outre, **la transition entre la phase de chauffage inerte et la combustion entretenue est approximative**. Dans les modèles qui considèrent un allumage instantané, la surface du propergol passe immédiatement d'un comportement de paroi inerte à une condition de paroi transpirante avec une température de flamme ( $\approx 3000$  K) et un débit massique ( $\approx 10$  kg/m<sup>2</sup>/s) élevés, et il est raisonnable de se demander si une transition aussi abrupte est physiquement acceptable ou non, et quel est l'impact de la dynamique manquante sur le processus d'allumage global.

En ce qui concerne la question de la température d'allumage, des approches utilisant un modèle quasi-stationnaire de la flamme de propergol ont été développées à l'ONERA [51, 52, 61]. Une

expression de la contribution de cette flamme à l'échauffement du propergol est obtenue de manière analytique en supposant une distribution du taux de réaction de type flamme mince [27]. L'allumage se produit donc naturellement au fur et à mesure que la flamme se développe, sans recourir à un critère de température. Cependant, même si le passage à l'état allumé est plus progressif, il y a toujours une transition pendant laquelle la seule source externe de chaleur pour le propergol passe de la phase gazeuse représentée dans le solveur CFD (via le transfert de chaleur conjugué en surface) à la flamme du propergol représentée dans le solveur unidimensionnel du propergol, qui agit comme une barrière, isolant la surface du propergol de l'écoulement environnant. Cette transition et ses effets n'ont pas été étudiés jusqu'à présent. De plus, l'approche flamme mince pour est restrictive et n'est pas adaptée à certains propergols, en particulier les propergols dits nouveaux, qui comprennent des ingrédients plus sensibles comme le HMX [62, 63].

Ainsi, il est devenu évident qu'un modèle d'allumage plus générique est nécessaire, où la flamme du propergol est résolue numériquement sans hypothèses fortes telles que celle de flamme mince. Idéalement, la flamme du propergol devrait être résolue dans le solveur CFD, de sorte que toutes les interactions avec l'écoulement environnant puissent être représentées. Cependant, la combustion des propergols solides est généralement caractérisée par un mécanisme de réaction rapide et raide qui se traduit par une flamme très fine attachée à la surface du propergol, avec une épaisseur de l'ordre de 100  $\mu\text{m}$ . La résolution numérique d'une flamme aussi fine nécessiterait des mailles d'une épaisseur d'environ 1  $\mu\text{m}$  près de la surface du propergol. Il serait difficile d'utiliser un tel raffinement sur toute la longueur du moteur, qui peut atteindre 10 mètres ou plus. C'est la raison pour laquelle toutes les simulations d'allumage mentionnées précédemment utilisent des modèles simplifiés pour représenter la combustion du propergol, sans la modéliser à l'intérieur du modèle fluide du solveur CFD. Ceci réduit considérablement le raffinement du maillage nécessaire près de la surface. Cela permet aussi de ne pas prendre en compte dans le solveur CFD les réactions chimiques et les espèces gazeuses qui interviennent uniquement dans la flamme du propergol. Cette représentation simplifiée est utilisée localement, en chaque facette limite du maillage CFD appartenant à la surface de la charge de propergol.

Suite à notre remarque précédente sur le manque de généralité de ces modèles simplifiés, nous proposons dans ce manuscrit de les remplacer par un modèle plus détaillé. Une première idée serait d'utiliser un des modèles unidimensionnels détaillés présentés précédemment, qui tentent d'inclure tous les phénomènes physico-chimiques pertinents pour la combustion des propergols. Cependant ces modèles, même dans le contexte unidimensionnel, restent très coûteux, et une paramétrisation correcte de la cinétique de la phase gazeuse et de divers paramètres est encore incertaine en raison du manque de mesures expérimentales *in situ* précises et de la connaissance limitée de certains processus (couche liquide par exemple). Il semble donc plus raisonnable de simplifier la modélisation pour obtenir un modèle de complexité intermédiaire, plus facile à mettre en œuvre et à paramétrer pour différentes classes de propergols.

Cette thèse est motivée par l'idée qu'un tel modèle de l'allumage des propergols solides peut conduire à une plus grande précision dans le calcul des transitoires d'allumage, ainsi que permettre une représentation globale plus précise du couplage entre la dynamique de la chambre et la combustion du propergol solide en régime instationnaire.

Suite aux différents points évoqués dans cette introduction, la thèse s'est concentrée sur de multiples aspects, que nous récapitulons ici. Le premier est la **modélisation et la simulation unidimensionnelles de la combustion et de l'allumage des propergols solides**. Les points suivants devront être étudiés :

- choix d'un niveau de modélisation simple mais polyvalent, capable d'inclure plus de physique si nécessaire ;
- analyse du statut mathématique de la vitesse de régression de la surface du propergol, également appelé valeur propre, lorsqu'on considère une onde de combustion stationnaire ;
- analyse de la structure mathématique du système d'équations après semi-discrétisation en espace du modèle instationnaire ;
- développement d'une stratégie numérique d'ordre élevé en temps et à pas de temps adaptatif, ainsi que le code associé ;
- développement d'une stratégie d'optimisation pour déterminer les différents paramètres du modèle à partir des données expérimentales disponibles.

Le deuxième point principal est le **couplage du modèle unidimensionnel d'allumage du propergol avec un modèle multidimensionnel de l'écoulement dans la chambre de combustion** pour permettre la simulation de l'allumage des MPS. Les deux modèles sont simulés avec des solveurs séparés, qui doivent être couplés dans l'espace et le temps. Les points suivants doivent être abordés :

- définition du couplage en termes de flux échangés ;
- développement d'un algorithme de couplage temporel d'ordre 1 en temps ;
- exploration de techniques permettant une généralisation de ce couplage à des ordres supérieurs ;
- définition d'une configuration réaliste et discriminante pour la vérification ;
- étude de l'effet de l'hypothèse unidimensionnelle pour la flamme du propergol sur le transitoire d'allumage d'un moteur.

Le manuscrit est organisé comme suit. Le premier chapitre donne une introduction détaillée du sujet. Après un aperçu de l'histoire de la propulsion solide, ses principes physiques de base sont présentés. Les différentes familles de propergols solides sont décrites, et une analyse simplifiée donne un aperçu des performances des MPS. Ensuite, les principaux sujets de recherche active sont abordés, et l'intérêt de disposer d'outils de simulation haute-fidélité est mis en évidence. Un examen détaillé des stratégies de simulation de l'allumage à grande échelle dans un moteur est présenté, montrant en particulier la nécessité d'un modèle d'allumage unidimensionnel agissant comme une condition limite dynamique dans un solveur CFD.

La première contribution du présent manuscrit, présentée dans la partie 1, porte sur la simulation unidimensionnelle de la combustion des propergols solides. Le chapitre 2 présente la modélisation instationnaire choisie. Il est démontré que le cadre bas-Mach est pertinent pour la représentation de la flamme, et facilite divers aspects de la modélisation et de l'implémentation. Les spécificités liées aux propergols composites sont discutées, et l'approche ZN, qui offre une manière alternative de représenter les phénomènes en phase gazeuse, est examinée.

Dans le chapitre 3, un modèle simplifié est étudié en régime permanent. Il forme un problème non linéaire à valeur propre, où la valeur propre est la température de surface (ou de manière équivalente la vitesse de régression de la surface). Il se prête à une analyse mathématique approfondie. En étendant la théorie des flammes laminares [64] au contexte de la combustion des propergols solides, l'existence et l'unicité d'un profil de température et de la vitesse de régression associée pour l'onde de combustion stationnaire sont démontrées. Le raisonnement mathématique amène directement à la génération d'une méthode de tir, qui permet d'obtenir ce profil à la précision machine, et dont la convergence est assurée. Une telle méthode est très utile pour les études paramétriques et pour la vérification d'autres outils numériques. Cette contribution est une généralisation des modèles

en régime permanent antérieurs [62, 65], et fournit une vue intéressante sur le comportement de la valeur propre. Elle a fait l'objet d'une première publication dans le journal *Combustion Theory and Modelling* [66], et a été présentée à la conférence *EUCASS 2019* [67].

En utilisant la méthode des lignes avec une approche de volumes finis, nous montrons dans le chapitre 4 que la semi-discrétisation en espace des équations unidimensionnelles conduit à un système d'équations différentielles-algébriques (DAE) d'index 1, et que c'est également le cas pour la plupart des modèles de combustion unidimensionnels, même en dehors du cas particulier des propergols solides. Par rapport au chapitre 3, la valeur propre devient un vecteur, dont les composantes incluent la température de surface, mais aussi les valeurs discrètes du profil de vitesse de la phase gazeuse.

Le chapitre 5 décrit la recherche d'une stratégie numérique fiable, précise et efficace pour la résolution de la dynamique des propergols solides basée sur la modélisation précédente. L'intégration temporelle des DAE d'index 1 est examinée, en particulier les exigences spécifiques pour une résolution d'ordre élevé en temps. En particulier, un traitement original de l'équation de continuité conduit à une efficacité accrue par rapport aux approches traditionnelles. Des schémas spécifiques, connus sous le nom d'ESDIRK [68], offrent des propriétés intéressantes, parmi lesquelles un ordre élevé et des estimations d'erreurs pour l'adaptation du pas de temps. Ils sont sélectionnés pour être implémentés dans le nouveau code Fortran de combustion unidimensionnelle des propergols VULC1D. Le code est comparé en régime permanent avec l'outil semi-analytique précédent pour vérifier la discrétisation spatiale. La discrétisation temporelle est ensuite vérifiée en considérant la réponse du propergol à des oscillations de pression, pour laquelle il existe des solutions analytiques linéarisées. Enfin, l'ordre élevé de la convergence en temps est démontrée.

Dans le chapitre 6, trois cas tests exigeants sont présentés : allumage par flux laser avec une cinétique simplifiée, solution stationnaire linéairement instable conduisant à un cycle limite, et application d'une perturbation de pression à la solution stationnaire dans le cas d'une cinétique chimique détaillée. L'efficacité des méthodes adaptatives d'ordre élevé est clairement démontrée dans tous les cas, montrant une grande amélioration par rapport aux approches traditionnelles utilisant des méthodes d'ordre 1 ou 2.

Les chapitres 5 à 6 présentent des contributions clés pour la simulation unidimensionnelle de la combustion à faible nombre de Mach, que ce soit pour les propergols solides ou pour les flammes homogènes. Elles ont fait l'objet d'un second article [69], qui est actuellement en cours de révision pour le journal *Communications in Computational Physics*.

La deuxième partie traite de la stratégie utilisée pour coupler CEDRE et l'outil unidimensionnel. Le chapitre 7 présente le couplage en terme des variables et des flux échangés, et présente un algorithme de couplage en temps d'ordre 1. Une attention particulière est accordée à la formulation des flux échangés par les modèles de propergol et de fluide, à la conservativité globale, et à la gestion du flux de chaleur de paroi, en particulier pendant l'établissement de la flamme. Une variante du couplage est introduite, où la flamme est résolue dans le solveur CFD au lieu du solveur de propergol, permettant une approche de type simulation directe (DNS) qui peut être utilisée sur des configurations 2D simples pour étudier l'effet de diverses hypothèses, comme celle de l'unidimensionnalité de la flamme du propergol.

Dans le chapitre 8, nous montrons comment l'algorithme de couplage précédent peut être généralisé à un ordre supérieur via l'utilisation d'une extrapolation polynomiale en temps des variables de couplage. Ceci permet une adaptation dynamique de l'ordre et du pas de temps de couplage, afin d'améliorer la stabilité et la précision des simulations couplées. Un cas test simplifié illustre l'amélioration de la précision et de l'efficacité fournie par cette approche. La

méthode proposée peut facilement être appliquée à d'autres problèmes impliquant le couplage de deux solveurs via une interface physique, comme le transfert de chaleur conjugué.

La dernière partie du manuscrit concerne d'application du couplage du chapitre 7 à une configuration représentative d'un petit MPS, ainsi que les différents prérequis à la mise en œuvre de cette approche.

Le chapitre 9 montre comment les données expérimentales peuvent être utilisées pour ajuster les paramètres des modèles d'allumage. Une approche d'optimisation sous contrainte est proposée, qui garantit que les principales caractéristiques du propergol (loi de vitesse de combustion en régime permanent, température de surface physiquement acceptable) sont respectées, tout en maximisant l'accord entre les temps d'allumage laser simulés et ceux mesurés expérimentalement à l'ONERA.

Le chapitre 10 récapitule l'ensemble des développements de codes réalisés au cours de la thèse, qui constituent une part importante du travail de thèse. Les différents outils et codes sont décrits en termes d'objectifs, d'implémentation, de structure de données et autres détails.

Le chapitre 11 montre la conception d'un cas d'essai bidimensionnel, représentatif d'un petit MPS, pour simuler l'allumage avec les deux approches couplées du chapitre 7. Le cas test est conçu de manière à ce que toutes les échelles spatio-temporelles associées au modèle de l'écoulement puissent être résolues par le solveur CFD. Ainsi, les écarts entre les résultats des deux approches peuvent être directement attribués aux différences de modélisation de la flamme du propergol. Diverses simulations avec différents raffinements de maillage et pas de temps attestent de l'excellent accord des deux approches, ce qui démontre qu'une représentation unidimensionnelle de la flamme du propergol comme condition aux limites pour le code CFD est précise. Ce chapitre, ainsi que le chapitre 7, a fait l'objet d'un acte de conférence [70] et d'une présentation lors de la conférence *AIAA Propulsion and Energy Forum 2021*, et un article plus détaillé est en préparation.

Enfin, une conclusion est présentée pour résumer le travail accompli au cours de la thèse, et des perspectives pour de futures recherches sont présentées.



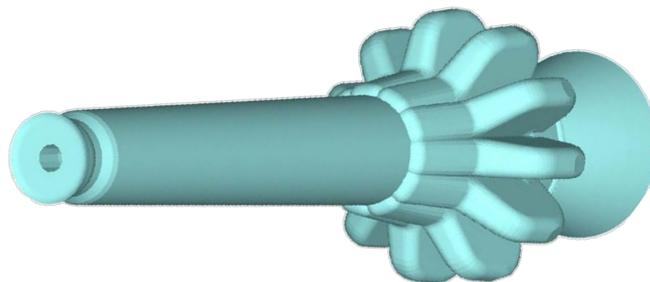
# General introduction

Rockets are one of the pinnacles of the technical progress achieved in the 20<sup>th</sup> century. They are now used routinely for multiple applications, ranging from space launchers for satellite delivery or human space flight, to weapon propulsion systems. The single most important element of a rocket is the motor, which generates the thrust required to accelerate the payload. Various rocket motor types are available, the major ones being liquid-fuel engines and solid rocket motors (SRMs) [1].

SRMs use a solid-state propellant, which is manufactured typically as a cylinder, in the middle of which an initial extrusion constitutes the combustion chamber. After an ignition process, the surface of the solid propellant decomposes and gasifies. The resulting gaseous products burn without any external injection of fuel or reactant. SRMs possess very valuable advantages over other systems:

- very large motors can be manufactured, which allow for very high levels of thrust to be generated, for instance up to 1200 tons for each booster of the American Space Shuttle;
- they are much simpler in terms of structure, with almost no moving parts;
- they can be stored over long periods of time without specific maintenance;
- they can be ignited rapidly on demand, without the complex propellant cooling process of liquid-fuel engines.

Thus, they are used on a number of heavy-lift launchers, e.g. Ariane 5, the upcoming Ariane 6 and SLS (Space Launch System), as well as low-cost smaller rockets, e.g. the Italian launcher Vega or small sounding rockets used for suborbital experiments. The previous advantages also make them particularly appropriate for the military, both for small weapon systems, such as hand-held rocket launchers, and conventional or long-range ballistic missiles.



**Figure 5** 3D view of the nozzle and initial propellant grain shape in the P80 motor of the Vega launcher (©ArianeGroup)

Since the 1950s, the search for improved performance and reliability has motivated extensive research. Experiments on full-scale motors are however expensive and time-consuming, and they cannot be used to perform parametric studies to investigate the sensitivity of a given motor to design parameters for instance. Simplified theoretical modelling of SRMs has brought a better understanding of various subjects, such as pressure instabilities [2, 3], ignition [4], internal geometry

evolution [5]. However actual motors are often too complex in terms of the number and nature of physical phenomena involved, and also the internal geometry of the propellant load (or grain), as seen for example in Figure 5. Consequently, their dynamics cannot be accurately represented or predicted with simple models. Therefore, numerical simulation of more complete models, the theoretical analysis of which is out of reach, has gained a lot of attention since the 1970s, when computing power and numerical methods became adequate.

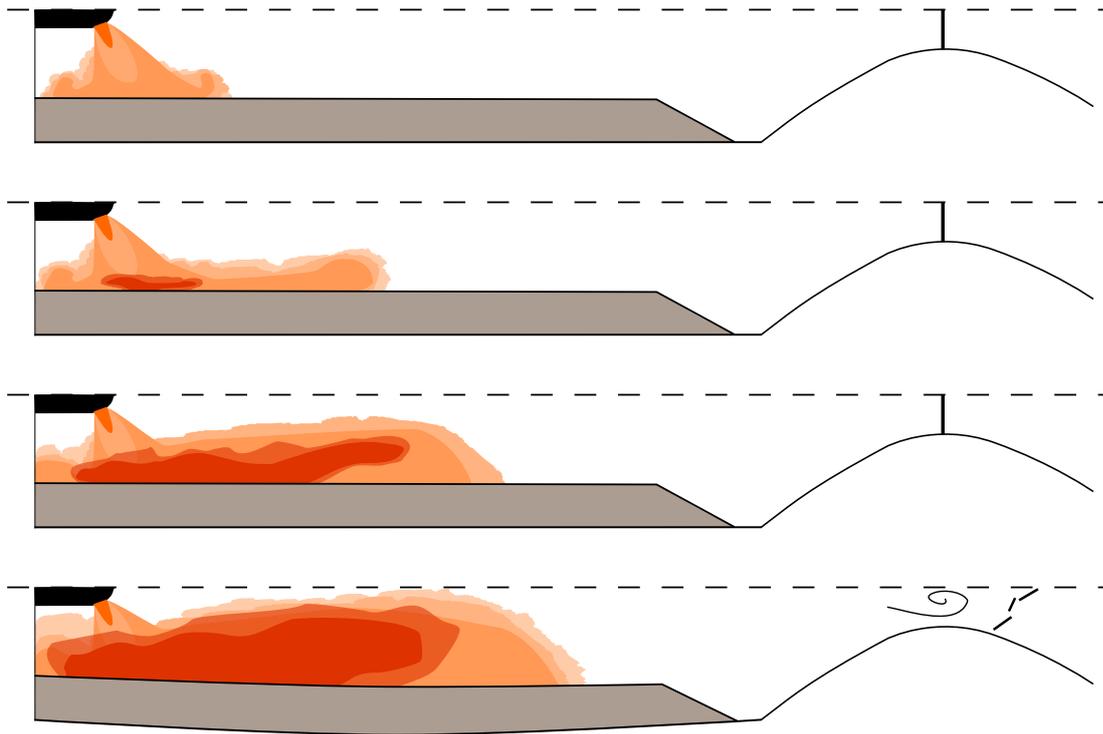
Now, a large number of simulation tools have been developed for SRMs for a variety of topics. For instance, the evolution of thrust during a complete firing can be accurately predicted [6]. In particular, tools allow for the prediction of the internal geometry during a firing, as the solid propellant is depleted [7, 8]. Instabilities of the chamber pressure linked to interactions between hydrodynamic effects, acoustics and combustion of the solid propellant have been investigated primarily via simulation [9–11]. Effects of the multiphase character of the flow inside the combustion chamber have led to a large body of numerical work [11–15], and much research is still devoted to this topic. The multiphysical complexity of estimating the electromagnetic signature of SRM plumes has also been investigated [16]. To improve on the performance of current propellant formulations, detailed reaction mechanisms have also been investigated [17, 18], and simulations of the detailed microstructure of composite solid propellants (an important class of solid propellants, where the material is heterogeneous) are also available [19–21].

Thanks to the extensive research and development since the 1950s, solid rocket propulsion has become the most reliable high-thrust motor type, with almost no in-flight motor failure in the US and in Europe for the last twenty years [22]. In fact, the single failure due to a solid rocket motor during that period happened with the Vega VV-15 mission, where an undersized thermal protection of the second stage resulted in the explosion of the motor [23]. Even though the failure rate of SRM is low, a large number of physical phenomena are still poorly understood and predicted.

One particular subject of interest is that of ignition. This is the process of initiating the combustion of the propellant load. A solid propellant typically burns at surface temperatures in the range of 700-1000 K, however an SRM is resting at ambient temperature before a firing. Thus, the propellant surface must first be heated up. This is usually done by using a smaller motor (pyrogen igniter), to generate a large volume of hot ( $> 2000$  K) gases, which are expelled at high speed towards the main load. Non-uniform heating of the latter ensues and ignition occurs at a few points first, establishing a propellant flame which then spreads to the rest of the grain.

The ignition phase is critical, as the SRM experiences strong spatial and temporal gradients of pressure, temperature and velocity inside the combustion chamber. The evolution of pressure during that phase is a dimensioning factor for the mechanical structure of the SRM. To ensure a safe operation, this pressure should not exceed the design limits, so as not to cause a structural failure or damage the propellant load, impacting the SRM thrust levels after ignition.

This requires an adequate sizing of the igniter system, its charge and the energy it delivers. The large volume of hot gases generated by the igniter are usually directed towards the main load via small nozzles, the orientations of which should be appropriately chosen. The igniter system should ensure that complete ignition occurs rapidly, so as to avoid misfires which could lead to an unstoppable slow burning of the propellant. The rise in pressure during ignition should also be sufficiently gentle, so that the structural deformations do not lead to the formation of cracks within the propellant, to the rupture of some portions of the load, or to a poor adherence between the load and the motor casing, which could lead to the accumulation of hot gases that may cause an undesirable ignition and rupture of the casing following overpressure. Also, the SRM combustion

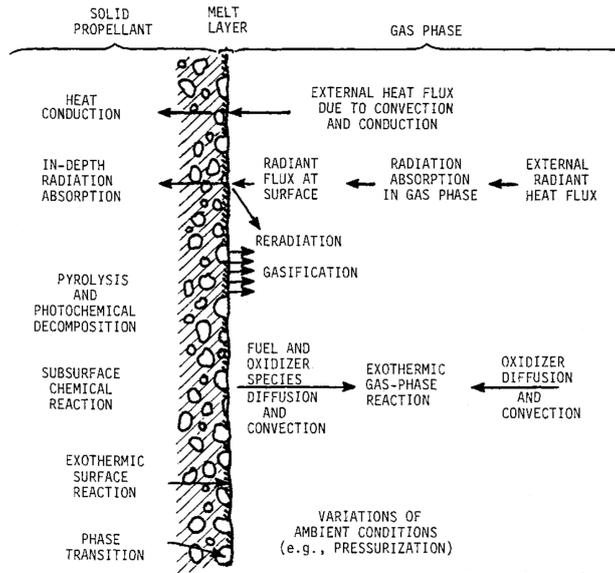


**Figure 6** Main phases of the ignition transient of an axisymmetric SRM. From top to bottom: igniter start-up and impingement, first ignition of the main load, flame spreading, mechanical deformation and rupture of the nozzle cap.

chamber is usually sealed before ignition. The sealing cap must be designed to break cleanly at a high enough pressure, so that the igniter gases remain within the combustion chamber, improving the energy transfer to the main load. The rupture should be clean and not lead to a partial obstruction of the nozzle, and the associated pressure oscillations should be well contained. Finally, the ignition of the main grain should be symmetrical, so as to avoid an uneven consumption of the propellant, which could lead to asymmetrical thrust at the end of the burn.

The various phases of an ignition transient are represented in Figure 6. The ignition transient involves a variety of phenomena, among which: heating of the unignited propellant through contact with the igniter flow, pyrolysis of the propellant at its surface (production of gaseous species), rapid combustion of the pyrolysis products in the immediate vicinity of the surface, acoustic and hydrodynamic phenomena inside the combustion chamber, multiphase flow (presence of liquid and solid particles), large range of flow velocities (1-1000 m/s), radiant emission from the droplet, flame and surfaces, deformation of the propellant load, rupture of the nozzle seal... All these phenomena contribute to the highly multi-physical and unsteady character of the solid rocket ignition and internal ballistics. Furthermore, ignition is highly dependent on the internal geometry of the propellant grain, as some portion of the propellant surface may be very little exposed to the flow of ignited gases. These aspects make it a formidable task to understand and predict the ignition process. It is of paramount importance that ignition be correctly characterised and, to improve the design process of an SRM, accurately predicted by numerical simulation tools.

An important ingredient of the prediction of an SRM ignition transient is the prediction of



**Figure 7** Phenomena involved in a solid propellant combustion wave [38]

the ignition of the solid propellant itself. Experimentally, this is conveniently studied with small propellant samples placed in a controlled environment and exposed to a hot flow [24] or a laser heat flux [25]. This allows for the determination of the ignition law of a propellant, i.e. the time it takes for the propellant to ignite when exposed to a constant heat flux at its surface.

However, the heat flux at the propellant surface in an SRM varies widely in space and time during the ignition transient. Therefore, a dynamic model is required to compute the ignition of the propellant in this case. For that purpose, it is convenient to consider a one-dimensional representation of the solid propellant in the direction perpendicular to the surface, as the various phenomena (heat conduction, conjugate heat transfer with the surrounding gas flow) occur predominantly along that direction. Early analytical ignition models [4, 26] relied on simplified representations of the various phenomena: constant heat flux, no modelling of the gas phase. Later models relied on a numerical resolution of the unsteady heat conduction within the solid and on simplified propellant flame models, either based on a high activation energy limit [27], or on a more generic but less physics-based model [28, 29], where the distribution of the gas-phase heat release is prescribed, instead of using a reactive flow model to solve for the associated flame profile. To circumvent the issue of the flame modelling, the Zeldovich-Novozhilov framework [30] discards the gas-phase part of the model, and replaces it by empirical relations deduced from experiments, however such data is usually imprecise and insufficient for a proper simulation of ignition without relying on additional models [31].

Some studies have been presented, which investigate the one-dimensional ignition of a solid propellant, using a detailed description of the propellant surface phenomena (e.g. presence of a small liquid layer) and complex gas-phase kinetics [18, 32–34]. These studies are usually generalisation of steady-state or unsteady simulation tools [18, 35–37]. All these tools solve a comprehensive set of equations: unsteady heat conduction in the solid propellant, in-depth degradation and decomposition of the propellant within the solid phase, melting of decomposition products in the surface liquid layer, evaporation of that layer and combustion of the evaporated products in the low-Mach gas phase. The large number of processes occurring in the propellant flame structure are summarised in Figure 7.

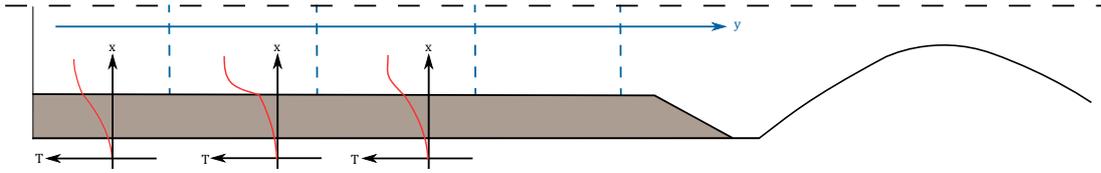
Note that little work has been focused on the study of ignition for a geometrically-detailed

model of heterogeneous solid propellants. Due to the computational cost, such simulations are only tractable with simplified global kinetics [21, 39]. The previous detailed one-dimensional tools are of great interest for studying the effects of fundamental physico-chemical processes. However, they are highly complex, and rely on a split integration of the gas and solid phases, with various integration algorithms of low-order for each subproblem. **The coupling between the various solvers is only first-order accurate in time.** Hence, the computational efficiency is not optimal, **the limited stability inhibits the use of highly-refined spatial meshes** [33]. Part of the issue comes from the **lack of careful analysis of the mathematical nature of the overall discrete system** to be solved. The latter is differential-algebraic of index 1, for which the first-order implicit temporal scheme backward Euler, often used in the one-dimensional solid propellant or low-Mach combustion communities, works well. However, **generalisation to higher orders has not been properly addressed yet.** Having a higher-order numerical strategy is of paramount importance to improve the computational efficiency and to better capture the dynamics of the simulated system. Indeed, the advantage of higher-order methods has been clearly identified in a variety of works for high-fidelity simulations of other dynamical systems [40, 41].

Let us come back the topic of simulating the ignition transient in SRMs. A strong research effort has been conducted in this direction since the 1950s. Early studies relied on simplified models that assumed a zero-dimensional representation of the combustion chamber [42–45]. A number of ad hoc parameters needed to be tuned to properly match experimental results, thus that approach could not be used in a predictive manner, in particular for long motors, where the pressure field is non-uniform across the chamber.

Around that time, numerical methods started to appear for the solution of the unsteady Euler equations, making it possible to solve the unsteady evolution of the compressible flow inside the motor, initially with quasi-one-dimensional CFD solvers [46], where the single space coordinate for the chamber flow model is along the length of the motor. Simulations accounting for spatial non-uniformities along the length of the motor (e.g. pressure) have enabled a large increase in the fidelity of the numerical results compared to experiments [46, 47]. To reproduce the progressive heating of the propellant, unsteady heat conduction in the solid is accounted for. Following the observation that heat conduction occurs predominantly in the direction perpendicular to the propellant surface [43], the heat conduction and propellant combustion modelling has been simplified to a one-dimensional approach. At each cell of the quasi-one-dimensional chamber flow spatial mesh, a one-dimensional heat equation is used to model the in-depth heating of the propellant in the direction perpendicular to the propellant surface, i.e. usually perpendicular to the longitudinal axis of the motor. Multiple instances of the heating model are used along the length of the motor to account for the propellant spatially non-uniform heating rate and temperature, as depicted in Figure 8.

This model could in theory be any of the previously discussed one-dimensional ignition models, which mostly differ by their representation of the propellant flame. However, to optimise the computational cost and to rely only on a simple parametrisation, the chosen ignition models are much simpler. The unsteady heat conduction in the solid is still solved as in the previous models, however the propellant flame is not accounted for. Instead, the propellant is assumed inert, as long as the surface temperature is below a user-specified ignition temperature. Once this value has been reached, the one-dimensional solid-phase heat conduction model is discarded, and it is assumed that the propellant is instantaneously ignited and that it burns in a quasi-steady manner with respect to the surrounding flow.



**Figure 8** Sketch of the one-dimensional ignition SRM model: the axis for the quasi-1D chamber flow is in blue, and the blue dashed vertical segments show the limits of the CFD cells discretising the chamber. The 1D propellant models are shown for the first 3 cells, and the red curves indicate the local propellant temperature profile.

Evolutions of such early one-dimensional SRM ignition modelling are still widely used today [48, 49], using more refined descriptions of the igniter flow impingement on the propellant surface and of the radiative heat transfer. Thanks to the simplicity of the models, parametric studies around known configurations can be performed at a low computational cost. However, such tools cannot be used to predict the behaviour of a completely new motor design. A large number of experiments is required to properly tune some of the model parameters, in particular with regard to heat transfer in zones where the propellant load is highly three-dimensional, as is the case in many SRMs.

Thus transition to 2D or 3D models of the combustion chamber flow field is necessary. Owing to the computational requirements of such simulations, the first studies only appeared in the 1990s [50–52]. To match the experimentally observed pressure evolution on various motors, radiative heat flux emanating from the propellant flame has been reported to be an important factor for a correct prediction of the flame spreading [52]. This makes certain areas, such as small gaps within segments of the propellant load, ignite much faster, since they are otherwise little exposed to the igniter flow. A large research and development programme has been conducted at CSAR (Center for Simulation of Advanced Rockets) in the 2000s has led to the development of ROCSTAR, a software suite able to simulate ignition of SRMs and account for the dynamic deformation of the propellant load [7, 53–55]. In all these works, the propellant ignition model remains the same as in the earlier SRM ignition studies. The solid propellant initially behaves as an inert material and, upon reaching a user-specified ignition surface temperature, instantaneously switches to a quasi-steady burning behaviour, only affected by the local wall pressure prescribed by the CFD solver. The propellant flame is not modelled itself, but the flame temperature is computed *a priori*, assuming chemical equilibrium. Similar tools have been developed recently in China [56–59] with the same ignition modelling.

These models suffer from several limitations. The first one is the reliance on a **constant ignition temperature criterion** to determine if the propellant is locally ignited. Actually, it has been shown that this temperature greatly depends on the surrounding conditions and on the wall heat flux history [27, 60]. A second limitation is that all reported ignition simulations couple the fluid flow and propellant solvers with **only first-order accuracy in time**, which could impact the precision of the solution and the computational efficiency. A third limitation is the **restrained accuracy of the propellant flame representation**. Indeed, from the point of view of the CFD solver, the flame is modelled as a surface phenomenon. In reality, it has a thickness on the order of 100  $\mu\text{m}$ . Compressing that flame into a surface phenomenon may discard some intricate physical interactions with the surrounding flow. Indeed, even though results reported in the literature obtained with this one-dimensional flame approach are usually in good agreement with experimental data, this does not constitute a complete verification of the physical coherence of this simplification.

Also, **the transition between the inert heating phase and the sustained combustion is approximate**. In the models that consider an instantaneous ignition, the propellant surface immediately switches from an inert wall behaviour to a transpiring wall condition with high flame temperature ( $\approx 3000$  K) and mass flow rate ( $\approx 10$  kg/m<sup>2</sup>/s), and it is sensible to wonder whether such an abrupt transition is physically acceptable, and what is the impact of the missing dynamics on the global ignition process.

Regarding the issue of the temperature criterion, ignition models that use a quasi-steady propellant flame have been developed at ONERA [51, 52, 61]. A formula for the contribution of this flame to the propellant heating is obtained analytically by assuming a flame-sheet reaction rate distribution [27]. Ignition thus occurs naturally as the flame develops, without relying on a temperature criterion. However, even the change to ignited state is smoother, there is still a transition during which the sole external source of heat for the propellant switches from the CFD gas phase (via the conjugate heat transfer) to the propellant flame, which is considered to isolate the propellant surface from the near-surface thermal profile found in the CFD domain. That transition and its effects have so far not been investigated. Furthermore, the flame-sheet approach for the propellant flame is restrictive and not suited to certain propellants, in particular so-called new propellants, which include more sensitive ingredients such as HMX [62, 63].

Thus, it has become apparent that **a more generic ignition modelling is needed**, where the propellant flame is solved numerically without strong assumptions such as the flame-sheet one. Ideally, the propellant flame should be captured within the CFD solver itself, so that all interactions with the surrounding flow can be represented. However, solid propellant combustion is usually characterised by a very stiff and rapid reaction mechanism that results in a very sharp flame attached to the propellant surface, with a thickness on the order of 100  $\mu\text{m}$ . Numerically resolving such a fine flame would require mesh cells with a thickness of around 1  $\mu\text{m}$  near the propellant surface. It would be intractable to use such a refinement across the whole motor length, which can be up to 10 meters or more. That is the reason why all the previously mentioned ignition studies use simplified models to represent the propellant combustion without actually modelling it inside the CFD model, largely decreasing the near-surface mesh refinement needed, and discarding the kinetic reaction and species found in the propellant flame. This simplified representation is used locally at each propellant boundary face of the CFD mesh.

Following our previous remark on the lack of genericness of these simplified models, we propose in this manuscript to replace them by a more detailed model. A first idea would be to use one of the highly-detailed 1D models discussed previously, which attempt to include all physico-chemical phenomena relevant to propellant combustion. However, these models, even in the one-dimensional context, remain very computationally intensive, and a proper parametrisation of the gas-phase kinetics and various parameters is still uncertain due to the lack of precise *in situ* experimental measurements and scarce knowledge of some processes (liquid layer for instance). Thus, it seems more reasonable to simplify the modelling to obtain a much lighter model, easier to implement and to parametrise for different classes of propellants.

This PhD thesis is motivated by the idea that a more detailed solid propellant ignition model can lead to greater precision in the computation of ignition transients, as well as enable an overall more accurate depiction of the coupling between the combustion chamber dynamics and the solid propellant combustion in the unsteady regime.

Following our previous discussions, the thesis has focused on multiple aspects, which we recapitulate next. The first one is the **one-dimensional modelling and simulation of solid propellant**

**combustion and ignition.** The following points should be investigated:

- choice of a simple yet versatile modelling level, able to include more physics if necessary;
- analysis of the mathematical status of the regression rate, also referred to as the eigenvalue, when considering a steady travelling combustion wave;
- analysis of the mathematical structure of the system of equations after semi-discretisation in space of the unsteady model;
- development of a numerical strategy with high-order of convergence in time and adaptive time-stepping, and of the associated code;
- development of an optimisation strategy to determine the various model parameters based on experimental data.

The second main point is the **coupling of the one-dimensional propellant ignition model with a combustion chamber flow field model** to enable the simulation of SRM ignition. Both models are simulated with separate solvers, which must be coupled in space and time. The following points should be tackled:

- definition of the coupling in terms of fluxes exchanged;
- development of a first-order temporal coupling algorithm;
- work towards a generalisation of this coupling to higher orders;
- definition of a challenging and discriminating configuration for verification;
- investigation of the effect of the one-dimensional assumption of solid propellant combustion on the ignition transient of a motor.

The manuscript is organised as follows. The first chapter gives a detailed introduction of the subject. After an overview of the history of solid propulsion, the basic physical principles are presented. The different families of solid propellants are described, and a simplified analysis gives an insight into the performance of solid rocket motor. Next, the major topics of active research are discussed, and the interest in having high-fidelity simulation tools is demonstrated. A detailed review of the strategies for the simulation of ignition at large-scale in a motor is conducted, showing in particular the need for a one-dimensional ignition model acting as a dynamic boundary condition in a CFD solver.

The first contribution of the present work, presented in Part 1, focuses on the one-dimensional simulation of solid propellant combustion. Chapter 2 presents the chosen unsteady modelling. The low-Mach framework is shown to be relevant for the gas-phase phenomena considered, while facilitating various aspects of the modelling and forthcoming numerical implementation. Specificities related to heterogeneous propellants are discussed, and the Russian Zeldovich-Novozhilov (ZN) approach is reviewed, which offers an alternative way of representing gas-phase phenomena.

In Chapter 3, a simplified model is studied in steady-state. It forms a nonlinear eigenvalue problem, where the eigenvalue is the surface temperature (or equivalently the surface regression speed), and the problem is amenable to an in-depth mathematical analysis. Extending the theory of laminar flames [64] to the context of solid propellant combustion, the existence and uniqueness of a solution profile is proved. The reasoning behind the proof can be used directly to construct a shooting method which allows to obtain this profile with machine accuracy following an iterative approach, the convergence of which is ensured. Such a method is very useful for parametric studies and for the verification of other numerical tools. This contribution is a generalisation of earlier steady-state models [62, 65], and provides an interesting view on the behaviour of the eigenvalue. It has been the subject of a first publication in the peer-reviewed journal *Combustion Theory and Modelling* [66], and has been presented at the *EUCASS 2019* conference [67].

Using the method of lines with a finite-volume approach, we show in Chapter 4 that the semi-

discretisation in space of the one-dimensional equations leads to a system of differential-algebraic equations (DAEs) of index 1, and that this is also the case for most one-dimensional combustion models, even outside the solid propellant particular case. Compared to Chapter 3, the eigenvalue becomes a vector, whose components include the surface temperature, but also the discrete values of the gas-phase velocity profile.

Chapter 5 describes the quest for a reliable, precise and efficient numerical strategy for the resolution of the solid propellant dynamics based on the previous modelling. Temporal integration of index-1 DAEs is reviewed, in particular the specific requirements for a high-order resolution in time of the fully-coupled system. In particular, an original handling of the continuity equation is proposed and yields an improved efficiency over traditional approaches. Specific schemes, known as ESDIRK [68], offer interesting properties, e.g. high-order and error estimates for time step adaptation. They are selected for implementation in the new one-dimensional propellant combustion Fortran code VULC1D. The code is compared in steady-state against the previous semi-analytical tool to verify the spatial discretisation. The temporal discretisation is then verified by considering the response of the propellant to pressure oscillations, for which there exists linearised analytical solutions. Finally, high-order of convergence in time is demonstrated.

In Chapter 6, three challenging test cases are presented: laser-induced ignition with simplified kinetics, linearly-unstable steady-state solution leading to a limit-cycle, and pressure perturbation with detailed kinetics. The efficiency of high-order adaptive methods is clearly demonstrated in all cases, showing a vast improvement over traditional low-order approaches.

Chapters 5 to 6 present key contributions for the one-dimensional simulation of low-Mach combustion, either for solid propellants or for homogeneous or spray flames. They have been the subject of a second article [69], which is currently under review for the journal *Communications in Computational Physics*.

The second part deals with the strategy used to couple the CFD toolchain CEDRE from ONERA [71] and the one-dimensional tool VULC1D. Chapter 7 presents the initial coupling, with first-order accuracy in time. Specific attention is given to the formulation of the fluxes exchanged by the propellant and fluid models, the overall conservativity, and the handling of the wall heat flux, in particular during the flame establishment. A variant of the coupling is introduced, where the flame is solved within the CFD solver instead of the propellant solver, allowing for a DNS approach that can be used on simple 2D configurations to study the effect of various assumptions, such as that of one-dimensionality for the propellant flame.

In Chapter 8, we show how the previous coupling algorithm can be generalised to higher-order via the use of polynomial extrapolation in time of the coupling variables. This enables a dynamic adaptation of the order of accuracy and of the time step, to improve the stability and the precision of the coupled simulations. A simplified test case illustrates the accuracy and efficiency enhancement provided by this approach. The proposed method can easily be applied to other problems involving coupling of two solvers via a physical interface, such as conjugate heat transfer.

The final part of the manuscript is concerned with the application of the coupled simulation framework to a realistic motor configuration.

Chapter 9 shows how experimental data can be used to fit the parameters of the ignition models. A constrained optimisation approach is proposed, which ensures the main characteristics of the propellant (steady-state burning rate law, physically acceptable surface temperature) are met, while maximising the fit of simulated laser-induced ignition times compared to experiments conducted at ONERA.

Chapter 10 recapitulates all the code developments conducted during the thesis, which form

the backbone of the original contribution of the present work. The different tools and codes are described in terms of purpose, implementation, data structure and other details.

Chapter 11 shows the design of a 2D test case, representative of a small rocket motor, to simulate ignition with both coupled approaches from Chapter 7. The case design makes it possible to resolve all flow scales in space and time, so that differences between the two approaches can be directly attributed to the discrepancies in the modelling of the propellant flame. Various simulations with different mesh refinements and time steps show the excellent agreement of both approaches, demonstrating that a one-dimensional representation of the propellant flame as a boundary phenomenon for the CFD code is accurate. This chapter, along with Chapter 7, has been the subject of a conference proceeding [70] and presentation at the 2021 *AIAA Propulsion and Energy Forum*, and a more detailed paper is in preparation.

Last, a conclusion is presented to summarise the work accomplished during the thesis, and perspectives for future research are presented.

# Chapter 1

## Background on solid rocket motors

In this introductory chapter, the history and basics of rocketry and solid propulsion are exposed. The scope is gradually refined to the framework of solid rocket motors (SRMs), presenting the involved physical phenomena and the existing theoretical, experimental and numerical work.

A solid propellant is a solid-state mixture of fuel and oxidizer. When heated up, the propellant eventually releases gases that burn in an anaerobic manner, i.e. without requiring any external injection of oxidizer. The flame, across which these gases burn, maintains the surface of the propellant at high temperature, so that combustion is self-sustained. A solid propellant burns in a controlled manner (deflagration), as opposed to explosives, which detonate.

Its combustion releases a large volume of hot gases. If those are expelled within a closed chamber or casing, the inner pressure rises rapidly. This is the basic principle of guns, where one of the part of the casing is actually the rear surface of a projectile. For rocket applications, the combustion chamber is opened onto the outer atmosphere via a nozzle which transforms the high internal energy (pressure and temperature) into a high exhaust velocity at the exit of the nozzle, creating a propulsive force.

### 1.1 Short history and current overview of solid propulsion

Solid propellant and solid rockets have a long history, dating back to the Middle Ages in Asia. The technology underwent tremendous developments in the 20<sup>th</sup> century. In the following section, we briefly review the main historical milestones, as well as the military and civil applications.

#### 1.1.1 Origins

The first reported use of solid propellants for propulsion is thought to occur in medieval China between the 10<sup>th</sup> and 13<sup>th</sup> century. More precisely, there are reports of the army of the Song dynasty using arrows propelled by gunpowder to push back the invading Mongols during the siege of Kaifeng in 1232 [72].

The technology started to spread in the Middle East during the following century and then reached Europe, where gunpowder weapons were used in the naval battle of Chioggia between the Republics of Genoa and Venice in 1380. It is believed these weapons were highly inaccurate and thus only used for the bombing of fortifications [73].

The next notable improvement was the appearance of cast iron rocket tubes in the Kingdom of Mysore in Southern India in the 1750s. These Mysorean rockets were roughly 20 cm long with

a diameter of 7 cm, holding a 500 g black powder charge, which could propel them up to 1 km. These rockets proved highly effective against the British troops during the Second, Third and Fourth Anglo-Mysorean Wars (1780 to 1799). Following their defeat, the British took interest in the technology and managed to take hold of some of these rockets. A military rocket programme was started in 1801 at the Royal Woolwich Arsenal under the supervision of William Congreve, resulting in the development of the Congreve rockets, though birth of breached-loaded cannons relegated the use of rockets to peacetime uses, such as fireworks. A detailed review of solid propellant history can be found in [74].

### 1.1.2 Military applications

War has been a major vector of progress, in particular World War II [75], where solid rockets started to be used massively in warfare. A notorious example is the truck-based rocket launcher the Katyusha BM-13 shown in Figure 1.1a. Its rockets used a double-base propellant and were prone to unstable combustion, which motivated the pioneering work of Zeldovich from 1942 on solid propellant combustion dynamics and instabilities [2]. Rockets were used extensively as air-to-ground and air-to-air weapons, such as the supersonic unguided rocket R4M used by the German Luftwaffe on the Me-262 jet engine fighter. Note that during World War One already, rockets designed by the French officer Le Prieur were used in air-to-air combats to destroy observation balloons.

Such were the first steps towards guided missiles, the development of which started near the end of the war. For this application, solid propulsion offered interesting advantages compared to liquid-fuel rocket engines. Solid propellant is easy to store over a long time without maintenance, and its operation and manufacturing are easier and cheaper. The first surface-to-air missile to enter active service, the American Nike Ajax, was introduced in 1954 with a solid propellant initial booster, and a liquid-fuel engine as sustainer.

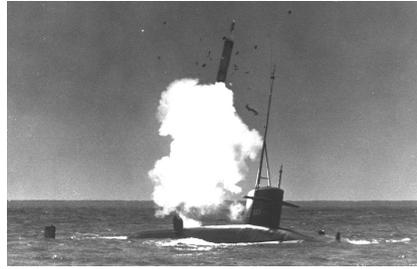
The first ballistic missile, the German V-2, used a liquid-fuel engine, inspiring the first post-war generation of ballistic missiles. The Cold War motivated vast programmes of research to produce such missiles with the ability to carry a nuclear warhead over several thousand kilometers. The USA originally intended to use their ground-based PGM-19 Jupiter liquid-fuelled missiles as their main Submarine Launched Ballistic Missile (SLBM). However using liquid fuel for submarines was inconvenient, as these had to be fed in shortly before firing, adding to the danger of the operation. Thus, the USA transitioned to SRMs in 1955 with the development of the Polaris A1 SLBM which, along with its subsequent variants, constituted the main submarine launcher for the USA until 1981 and Great Britain until 1996. A surface launch is shown in Figure 1.1b. For ground-based launches, the American Minuteman solid propellant missile was introduced in 1962 and is still deployed today. Most current nuclear ballistic missiles use solid-propellant motors, in particular the family of the M51 SLBM in France (see Figure 1.1c).

Today, solid rockets are widely used for hand-held rocket launchers, the majority of conventional missiles (air-to-air, surface-to-surface...) and ballistic missiles. Solid propellant is also used as a means of reducing drag for artillery shells, by improving the aerodynamics of the rear-end via mass injection, a technique known as base bleed. This widespread use of solid propulsion is due to several advantages of solid rockets over other propulsion modes. SRMs can be manufactured in large numbers easily, and offer good performances both for small and large motor dimensions. Most importantly, the propellant loading can be stored over long periods of time with little maintenance, and its initiation can be simply triggered, without any prior preparation process, thus enabling a fast intervention.

The interested reader is referred to [76] for more in-depth presentations by A. Davenas on the history of solid propellant development in the 20<sup>th</sup> century, and to [77] for the development of French solid rocketry, with a focus on military purposes.



(a) Katyusha BM-13 in 1942



(b) Polaris test launch in 1964



(c) M51 test launch (©Marine Nationale)

**Figure 1.1** Notorious examples of solid rockets used in the military

### 1.1.3 Space flight

Parallel to the ambitious military developments, space launchers also benefited from solid rocket technologies. In this field, SRMs have been primarily used as strap-on boosters to provide most of the thrust during the initial ascent, for example for the Atlas and Space Shuttle launchers in the USA, or Ariane 4 and 5 in Europe (see Figure 1.2a). The European launcher Vega-C (see Figure 1.2b) uses SRMs for its first three stages, and a liquid-fuel engine for the last one. The first stage is the same as the booster intended for use on the upcoming Ariane 6 launcher. The future launcher system SLS developed in the USA will also use strap-on boosters, which are an evolution of those used on the Space Shuttle. The manufacturing of a solid propellant motor being less expensive and complex than that of a liquid-fuel engine, it may be a more cost-effective choice, even though available data is very scarce on this financial aspect. However, the initial development cost is very high, requiring large research programs that cannot be afforded by smaller countries. Also, SRMs are not compatible with reusability, which is becoming the new trend for space launchers.

SRMs possess several advantages over competing liquid-fuel engines. First, they are relatively easy to design and manufacture, as they are made of only a few parts, which is in strong contrast with the turbopumps required by liquid engines. The only moving part is usually the nozzle, so as to allow for thrust vectoring. SRMs offer slightly lower performance but can easily be scaled up to produce very large amounts of thrust with a single motor, up to 1200 tons for one Space Shuttle booster (with a 500 tons propellant load).

One disadvantage of using SRMs lies in their relative inability to modulate thrust during the firing. Indeed, the thrust evolution in time is mainly dictated by the initial shape of the propellant grain. Due to the operating and manufacturing conditions, the overall trajectory cannot be precisely predicted, resulting in a certain imprecision for the insertion into orbit. That is one of the reasons why the last stages of all space launchers are propelled by a liquid-fuel engine, which can correct the deviation from the desired trajectory. A further advantage of having a liquid-fuelled upper stage is the ability to ignite the engine several times and perform multiple coast phases, enabling greater trajectory optimisation. For instance, the latest upper-stage engine Fregat from the Russian Soyuz rocket can be reignited up to 20 times.

For SRMs, thrust modulation techniques have been proposed, such as using a moving pintle to dynamically adjust the nozzle throat area [78] and therefore act on the internal chamber pressure as seen further in Equation (1.10). These have however not been used on launcher systems, because they may cause instabilities and pressures spikes [79–81]. Another method is the use of multiple propellant loads of different natures, that are consumed in sequence. Adjusting the time of ignition for the successive loads provides greater control on the overall thrust profile of the complete firing. These motors are traditionally referred to as dual-pulse and are mainly used for tactical missiles [82], because they allow to split the flight in multiple acceleration and deceleration phases, lowering the overall energy lost to drag. This results in a higher overall efficiency compared to a single-pulse motor, and a longer range.



(a) Ariane 5



(b) Vega

**Figure 1.2** Examples of civil launchers using SRMs

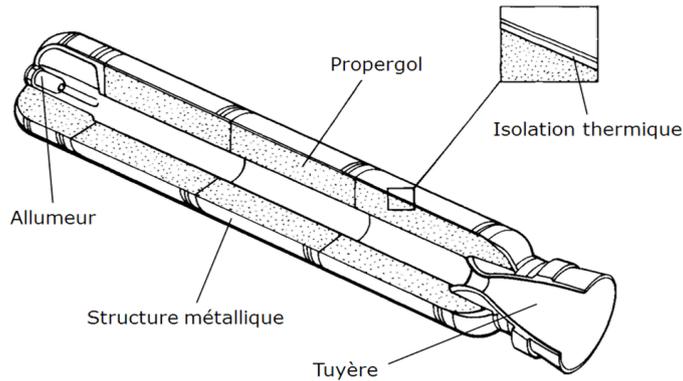
## 1.2 Rocket ballistics

In this section, we present a simplified description of a solid rocket motor, showing the link between its performance, interior dynamics and the characteristics of the solid propellant. The discussion is inspired by several reference books and courses, for instance [1, 6, 83].

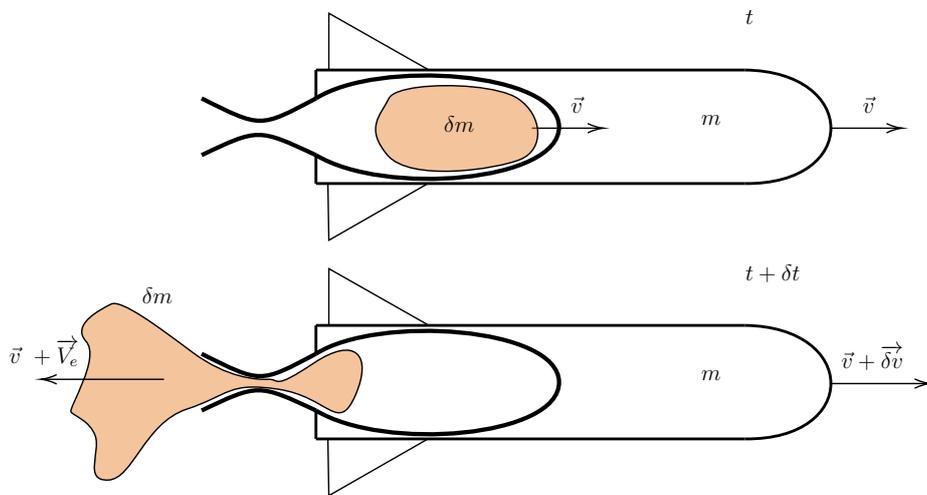
### 1.2.1 Structure of a solid rocket motor

A typical SRM is depicted in Figure 1.3. The main propellant load has a cylindrical shape with an extrusion at the center, which forms the combustion chamber. The propellant is often cast directly in the desired shape, and sometimes the cylinder is segmented in multiple parts along its length to facilitate manufacturing, in particular for large-scale motors. An insulating material isolates the propellant loading from the outer casing to protect the latter from strong thermal loads. It also improves the mechanical bonding between the grain and the case.

To ensure an efficient ignition of the motor, an igniter system is generally mounted at the forehead of the chamber. It is usually a small propellant load itself, which is initiated electrically. Its hot combustion gases are expelled towards the surface of the main propellant grain. The latter is locally heated up, and eventually ignites. The flame then spreads to the rest of the chamber. The combustion gases accumulate in the chamber and are expelled through a nozzle which converts their high internal energy (gas temperature between 2500 and 3500 K) into kinetic energy. The gases coming out of the nozzle are consequently at high velocity (2500 m/s for the Space Shuttle rocket boosters), generating an important thrust as will be shown via Equation (1.3).



**Figure 1.3** Cutaway of a typical solid rocket motor [6]



**Figure 1.4** Conservation of momentum for an idealised rocket

Usually, to improve the rapidity and reliability of the ignition phase, a nozzle cap initially seals the throat of the nozzle. This allows for the igniter gases to be contained within the chamber, increasing the thermal energy transferred to the propellant. The combustion products of the igniter and of the ignited parts of the main load accumulate, leading to a rapid rise in pressure. The cap ruptures once the combustion chamber reaches a sufficiently high pressure, and the nozzle throat is very quickly choked.

### 1.2.2 Propulsion principle

A rocket motor uses Newton's third law to propel a vehicle: taking reactants initially stored in the rocket, it generates a high-velocity flow of combustion gases that is ejected in a specific direction. This generates a propulsive force in the opposite direction, which can be determined by applying the conservation of momentum on the system formed by the rocket and the combustion gases flowing out of the nozzle. Let us assume  $m(t)$  is the mass of the rocket at time  $t$ , and  $\vec{v}(t)$  its speed. The gases are expelled from the nozzle with a velocity  $\vec{V}_e$  relative to the rocket. During an infinitesimal time interval  $\delta t$ , a mass  $\delta m = \delta t d_t m$  of gases is expelled from the rocket, with  $d_t m$  the mass flow rate exiting the nozzle. The situation is depicted in Figure 1.4.

Thus, neglecting gravity, pressure and aerodynamic forces, conservation of momentum leads to:

$$\underbrace{m(t)\vec{v}(t)}_{\text{total momentum at time } t} = \underbrace{(m(t) - \partial_t m \delta t)\vec{v}(t + \delta t)}_{\text{rocket momentum at time } t+\delta t} + \underbrace{\partial_t m \delta t(\vec{v}(t) + \vec{V}_e)}_{\text{momentum of the gas expelled during } \delta t} \quad (1.1)$$

$$m(t) \frac{\vec{v}(t) - \vec{v}(t + \delta t)}{\delta t} = -\partial_t m \vec{v}(t + \delta t) + \partial_t m(\vec{v}(t) + \vec{V}_e) \quad (1.2)$$

In the limit  $\delta t \rightarrow 0$ , we obtain:

$$m \partial_t \vec{v} = -\partial_t m \vec{V}_e \quad (1.3)$$

This equation and its resolution was presented in 1903 by the Russian scientist Tsiolkovski [84]. Let us consider the case where the rocket moves along  $\vec{x}$  towards  $x > 0$ , and the combustion gases are ejected in the opposite direction. We project the previous vector equation on  $\vec{x}$  and introduce  $V_e = \|\vec{V}_e\|$ . Let us consider a rocket launched with an initial mass  $m_i$ , which is the sum of its dry mass  $m_{empty}$  and its fuel load  $m_{fuel}$ . Integrating in time the previous equation yields the velocity increment achievable after having burnt all the fuel:

$$\Delta v = v_f - v_i = V_e \log \left( \frac{m_{empty} + m_{fuel}}{m_{empty}} \right) := V_e \log(s) \quad (1.4)$$

with  $s$  the structural coefficient of the rocket.

The velocity delta  $\Delta v$  increases as  $s$  increases ( $\frac{m_{fuel}}{m_{empty}}$  increases), and as  $V_e$  increases, i.e. if the gases are expelled faster. Therefore, a key point for the mechanical design of a space launcher is to have the highest structural coefficient, i.e. as little dry mass as possible. On the other hand, the rocket engine must maximise the ejection velocity. Let us study this point in more details.

### 1.2.3 Motor performance

A rocket motor aims at generating a high-velocity flow of gases. For this purpose, it is composed of a chamber, where the combustion of reactants (coming from the decomposition of the solid propellant for SRMs, or from separate reactant injectors for liquid-fuel engines) causes the chamber temperature to rise. The aft end of the combustion chamber connects to a nozzle which opens to the outer atmosphere. Due to its shape and limited cross-section, the nozzle limits the mass flow rate of gases that is expelled, thus causing the chamber to accumulate hot gases, resulting in a pressurisation of the chamber. If the motor is stable, an equilibrium point can be found such that the pressure does not diverge (explosion or extinction). The evolution of the chamber pressure and the position of the equilibrium point can be found by studying the *internal ballistics* of the motor, as described in the following.

#### Combustion of the solid propellant load

A first step towards the characterisation of an SRM is to study how the propellant load burns once ignition has occurred. A simulated regression of a 2D propellant load is shown in Figure 1.6. At a given time, the solid propellant surface burns in such a manner that, locally at any point of the surface, the regression velocity vector is perpendicular to the surface. The norm of this vector is the regression speed  $r$ , or burning rate. While it is complex to build a physics-based model for the evolution of  $r$ , numerous experiments have shown that, in steady-state,  $r$  can be well approximated by the following Vieille law:

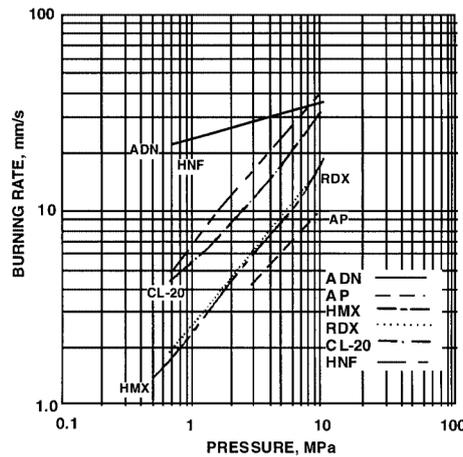
$$r = a P_c^n \quad (1.5)$$

with  $P_c$  the combustion chamber pressure, and the empirical parameters  $a$  and  $n$ , the latter being referred to as the *pressure exponent*. These parameters can usually be considered constant over a relatively large range of pressures, as seen for example from the linear relation between  $\log(P)$  and  $\log(v)$  in Figure 1.5 for monopropellants. Note that  $n$  can vary with  $P$ . In particular, some propellants exhibit a pressure range where  $n = 0$ , i.e. the burn rate is pressure-independent. This property is especially interesting for military applications, to ensure a good reproducibility of a firing under a wide variety of conditions. Note that  $a$  typically depends on the initial temperature of the propellant. The following form is often employed:

$$a = a_0 \exp(\sigma_T(T_0 - T_r)) \quad (1.6)$$

with  $a_0$  a constant,  $T_r$  a reference temperature, and  $\sigma_T = \partial_{T_0} \log(r)$ . Most composite propellants have their pressure exponents  $n \in [0.2, 0.5]$ , while typical homogeneous propellant have  $n \approx 0.7$ . The temperature sensitivity  $\sigma_T$  is usually on the order of  $10^{-3}$ .

For a complete combustion chamber, the instantaneous mass flow rate of solid propellant that is transformed into combustion gases is  $m_p = \rho_c A_b r$ , with  $\rho_c$  the density of the solid propellant and  $A_b$  the burning area, i.e. the solid propellant area exposed to the combustion chamber. This surface evolves with time, as the propellant load is consumed.



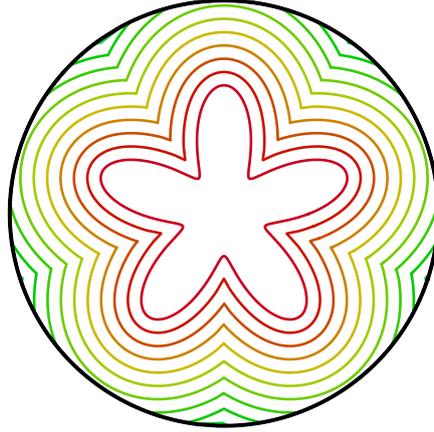
**Figure 1.5** Measured burning rates for six monopropellants [85]

Under atmospheric pressure, the typical regression speed is 1 to 10 mm/s. The gaseous products typically have a density 1000 times lower than the solid propellant, therefore their velocity at the end of the propellant flame is on the order of 1 to 10 m/s.

### Flow through the nozzle

The nozzle is a key element of the motor. Its role is to convert the high enthalpy of the combustion chamber gases into kinetic energy, i.e. it accelerates the combustion gases so as to provide the highest value of the ejection velocity  $V_e$ . Through the nozzle, the pressure drops from  $P_c$  in the chamber to  $P_e$  at the nozzle exit. Conversely, the flow speed increases up to the exit velocity  $V_e$  relative to the rocket. Assuming inviscid isentropic ideal gas flow in the nozzle, this velocity reads:

$$V_e = \sqrt{\frac{2\gamma}{\gamma-1} \frac{R}{\mathcal{M}} T_c \left( 1 - \left( \frac{P_e}{P_c} \right)^{\frac{\gamma-1}{\gamma}} \right)} \quad (1.7)$$



**Figure 1.6** Regression of a star-shaped solid propellant grain (own work [86])

with  $\gamma$  is the adiabatic index,  $T_c$  the chamber gas temperature,  $R$  the universal gas constant, and  $\mathcal{M}$  the average molecular weight of the combustion products.

The flow rate can be expressed as:

$$m_e = \frac{P_c A_t}{c^*} \quad (1.8)$$

with  $A_t$  the throat area, and  $c^*$  the characteristic speed defined as:

$$c^* = \frac{\sqrt{\gamma R T_c / \mathcal{M}}}{\gamma \left( \frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{2(\gamma-1)}}} \quad (1.9)$$

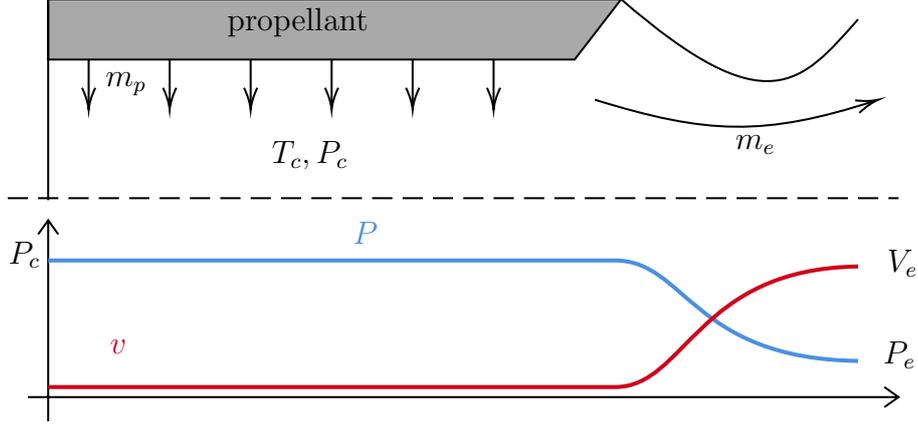
From the previous formula for the exhaust velocity, we see that the characteristic speed  $c^*$  is proportional to the maximum theoretical exhaust velocity, which is achieved if the outer pressure is 0. Therefore  $c^*$ , which is independent of the nozzle design, is a good indicator of the combustion chamber and propellant performance. From its expression, we see that having high-temperature combustion products with a low molar mass is beneficial.

## Chamber equilibrium

Knowing the behaviour of the solid propellant and the nozzle, the main aspects of the motor ballistics can be obtained with a simple model of the combustion chamber. Let us assume this chamber is zero-dimensional, i.e. the pressure and temperature fields are uniform. Let us further assume that the nozzle and the propellant have a quasi-steady behaviour, i.e. they instantly adapt to changes in the chamber pressure, and that the chamber temperature  $T_c$  is constant (equal to the final temperature of the propellant flame).

The internal flow field and mass fluxes are sketched in Figure 1.7. The volume of the combustion chamber is  $V$ . Writing the chamber mass balance, we can derive the following equation, which governs the temporal evolution of the combustion chamber pressure  $P_c$ :

$$\frac{V \mathcal{M}}{R T_c} \frac{dP_c}{dt} = m_p(P_c) - m_e(P_c) \quad (1.10)$$



**Figure 1.7** Simplified schematic of the internal flow of an SRM

This model is often referred to as *volume-filling approach* in the literature [60]. Note that the volume  $V$  of the chamber increases as the grain burns, but this happens very slowly compared to the characteristic time associated with pressure fluctuations in the chamber, therefore we can neglect this effect to study the stability of the motor. The equilibrium chamber pressure  $P_{c,eq}$  is such that:

$$m_e(P_{c,eq}) = m_p(P_{c,eq}) \Rightarrow \frac{P_{c,eq} A_t}{c^*} = A_b a P_{c,eq}^n \Rightarrow P_{c,eq} = \left( \frac{A_t}{A_b a c^*} \right)^{\frac{1}{n-1}} \quad (1.11)$$

The last expression is not valid if  $n = 1$ . In this scenario, both flow rates are linear in  $P_c$  and will not cancel each other: depending on the sign of  $A_b a - (A_t/c^*)$ , the pressure will rise or decay exponentially.

The equilibrium pressure is only stable if, locally,  $\partial_{P_c} RHS < 0$ , with  $RHS$  the right-hand side of Equation (1.10). In that situation, a perturbation that causes  $P_c$  to rise (respectively drop) will cause the right-hand side to drop (respectively rise), thus stabilising the pressure back onto its steady operating value:

$$\begin{aligned} \frac{\partial RHS}{\partial P_c}(P_{c,eq}) &= a(n-1)P_{c,eq}^{n-1} + \frac{A_t}{c^*} < 0 \\ \Leftrightarrow (1-n)P_{c,eq}^{n-1} &> \frac{A_t}{ac^*} \end{aligned}$$

All terms except  $(1-n)$  are positive, therefore the equilibrium pressure is only stable is  $(1-n)$  is positive as well, i.e. if  $n < 1$ . In practice, this is always the case.

## Thrust

Let  $P_{atm}$  be the atmospheric pressure surrounding the rocket. Under steady operating conditions, the overall thrust  $F$  produced by the engine is:

$$F = m_e V_e + (P_e - P_{atm}) A_e \quad (1.12)$$

with  $A_e$  the surface of the nozzle exit plane, and  $P_e$  the nozzle exit pressure.

It is useful to reformulate the thrust as:

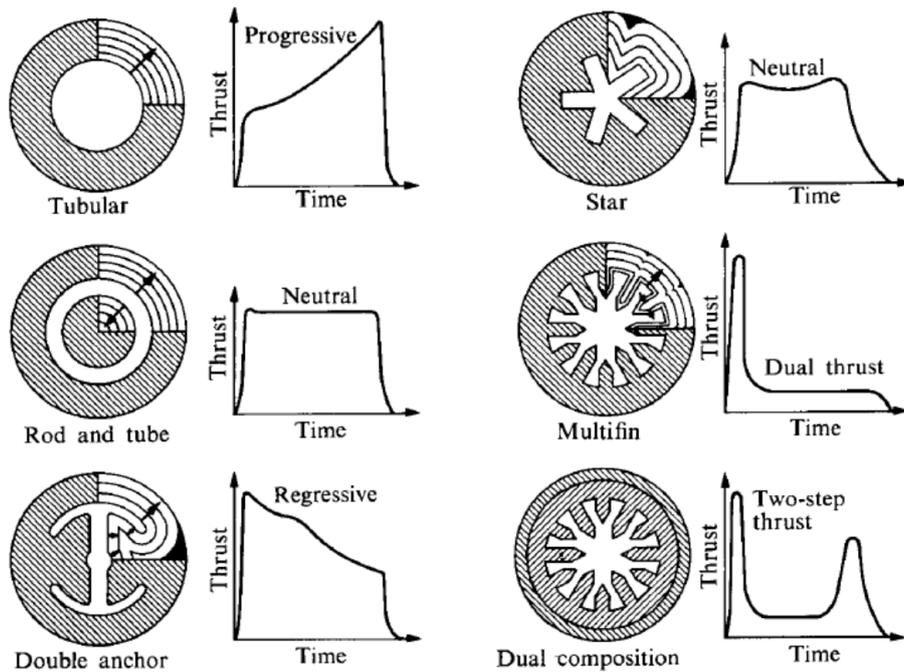
$$F = C_F P_c A_t \quad (1.13)$$

with  $C_F$  the dimensionless thrust coefficient, which can readily be expanded as:

$$C_F = \sqrt{\frac{2\gamma^2}{\gamma-1} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma-1}{\gamma}} \left[1 - \left(\frac{P_e}{P_{atm}}\right)^{\frac{\gamma-1}{\gamma}}\right] + \frac{P_e - P_{atm}}{P_c} \frac{A_e}{A_t}} \quad (1.14)$$

This coefficient only depends on the burnt gases (via  $\gamma$ ) and on the ratios  $A_e/A_t$  and  $P_c/P_{atm}$ .  $C_F$  increases as the nozzle efficiency increases, i.e. more thrust is produced with the same propellant and combustion process.

Once the propellant load of a motor has been ignited, the combustion is self-sustained and cannot be dynamically controlled. Following what we have exposed in the previous paragraphs, a means of imposing a certain thrust profile is to choose an adequate initial chamber shape. In Figure 1.8, we see that a wide variety of thrust profiles can be achieved by simply varying this initial shape. The shape can also evolve along the length of the motor. In fact, many motors have a star-shaped head end with a large number of branches (also called fins), so as to maximise the burning surface during the first phase of the flight. Other axisymmetric shapes are also possible, see Figure 1.9.

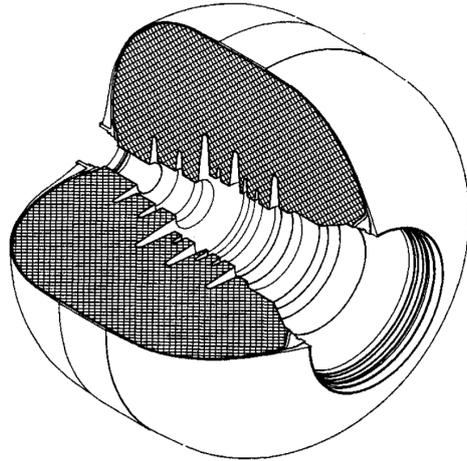


**Figure 1.8** Influence of the initial geometric configuration on the thrust profile during the firing [87]

## Specific impulse

To compare the efficiencies of different motors, a useful variable is the specific impulse  $I_s$ , defined as:

$$I_s = \frac{F}{g_0 m_{nozzle}} \quad (1.15)$$



**Figure 1.9** Cutaway of a typical second stage of an early French ballistic missile [77]

with  $g_0 \approx 9.81 \text{ m/s}^2$  the standard gravity acceleration per unit mass. Based on the previous equations, the specific impulse can be reformulated as:

$$I_s = \frac{C_F c^*}{g} \quad (1.16)$$

Conceptually, its value in seconds represents the duration during which 1 kg of fuel produces a thrust that is equal to the gravity force exerted on a 1 kg mass. Engines with higher specific impulse can produce a given thrust profile with less fuel than lower impulse motors, thus improving the overall performance as per Equation (1.4).

This form clearly shows that the overall performance of a motor is improved as the nozzle efficiency  $C_F$  and the characteristic speed  $c^*$  are increased. Previously, we have shown that the latter is proportional to  $\sqrt{T_c/\mathcal{M}}$ , hence increasing the combustion temperature  $T_c$  and lowering the mean molar mass  $\mathcal{M}$  of the combustion products benefit the performance. This motivates the use of metal particles and binder as additional fuel in composite propellants, as discussed in the next section.

For typical liquid-fuel engines, e.g. Vulcain on Ariane 5 or RS-25 for the Space Shuttle, the propellants are liquid oxygen and liquid hydrogen, which produce a final flame temperature of approximately 3500 K and combustion products with a mean molecular weight of 12 g/mol. Their overall specific impulse is close to 450 s, depending on the operating conditions (320 s at sea level for the  $H_2/O_2$  Vulcain engine on Ariane 5's main engine, and up to 434 s in vacuum). Liquid-fuel engines using kerosene and liquid oxygen, such the RD-107 on the Russian Soyuz and the Merlin 1D engine of SpaceX's Falcon 9 rocket, reach a much lower specific impulse of approximately 280 s at sea level (320s in vacuum).

Usual AP-HTPB-Al propellants yield combustion products that have a mean molecular weight of 30 g/mol and a temperature close to 3500 K. Solid rocket motors typically reach a specific impulse of between 250 and 280s (275s at sea-level for Ariane 5 SRMs). Thus, performance of SRMs is inferior. However their main advantage lies elsewhere: they are relatively cheap to build compared to the more complex liquid-fuel engines; they can be of very large size, producing large amounts of thrust. This makes them very appropriate as so-called “booster” stages for the initial ascent of a space launcher.

Note that a small increase in specific impulse can have a major impact on the overall performance

of the complete rocket system. For instance, sensitivity studies for a two-stage ballistic missile with a design range of 10 000 km show that an additional 1 second in specific impulse for both stages increases the range by 120 km [88]. Therefore, it is obvious why a number of research programs have focused on the development of higher-performance mixtures.

## 1.3 Families of solid propellants

In the previous section, we have seen the importance of solid propellant combustion (temperature and molar mass of the final products) on the overall performance of a rocket. We now shall take a look at the two main families of solid propellant (homogeneous and composite) and study their characteristics. Note that solid propellants may also exist in the form of gels, however these are not yet used extensively yet, will not be considered in this work.

### 1.3.1 Homogeneous propellants

A propellant is homogeneous when it is constituted of one or multiple ingredients, which are mixed at the molecular level. Each of these ingredients usually incorporate both the main oxidizing and the main reducing elements involved in the energy release during combustion. Most homogeneous propellants can be classified in the following categories [18]:

- Nitramines (RDX, HMX, HNIW, HNF)
- Azides (GAP, BAMO, AMMO)
- Nitrate esters (NG, NC, BTTN, TMETN, DEGDN)
- Nitrates (ADN, AN)

Such a propellant is produced by mixing the ingredients at temperatures between 30 and 100 °C [6], making use of the microscale interactions between the two ingredients. The high temperature facilitates the gelatinisation, which is the loss of the fibrous structure of nitrocellulose. Additives may be added to increase the long-term stability of the propellant (inhibiting slow decompositions during storage) and to tailor its ballistic properties (e.g. dependence of the burning rate on pressure).

Homogeneous propellants produced without solvent (used to enhance the initial gelatinisation) can only be extruded to form the final propellant grain. Hence they are limited to small loads: a few grams in small pyrotechnic systems, up to a few dozen kilos for the initial accelerator of some missiles. When a solvent is used, the propellant may be cast, allowing for more flexibility in the final shape and quantity, and productivity can be much higher, making the process adequate for systems produced in large amounts, such as anti-tank rockets.

In practice, homogeneous propellants have a very low temperature sensitivity, i.e. their combustion and ballistic behaviours are not affected much by a change in the propellant initial temperature before firing. Due to the low content in metallised additives and the absence of condensation of their combustion products, these propellants produce very little smoke. This is an important reason for their use in military applications such as hand-held rocket launchers.

### 1.3.2 Composite propellants

Composite propellants are made of solid powder oxidiser mixed within a polymeric matrix (binder) that governs most of the mechanical characteristics of the propellant load, and also acts as a secondary fuel. Most binders are inert, i.e. they cannot burn on their own. As opposed to the homogeneous propellant case, the oxidising or reducing elements are not held within a single molecule, but are juxtaposed in a composite structure. This structure greatly influences the various characteristic (mechanics, burning rate) of the propellant.

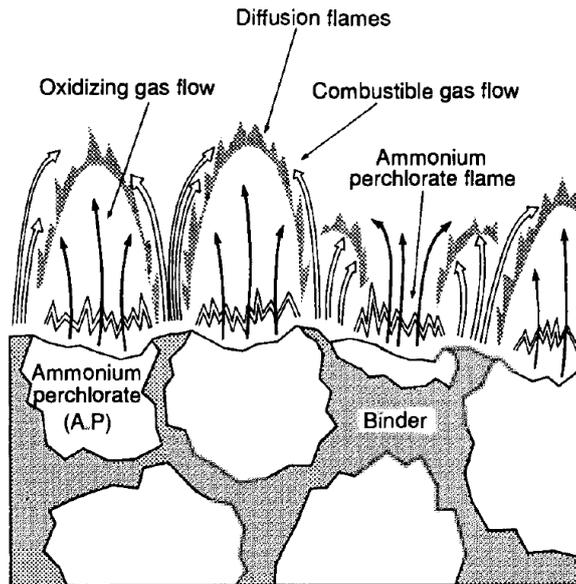
The binder is chosen such that it may hold as much oxidiser as possible while retaining good mechanical properties. From the 1950s up to today, much research has been conducted to find the optimal formulation (binder/oxidiser couple). Various trials with oxidisers such as ammonium perchlorate  $NH_4ClO_4$  (AP), ammonium nitrate  $NH_4NO_3$  (AN), potassium nitrate  $KNO_3$  (KN), and polymeric fuels such as hydroxyl-terminated polybutadien (HTPB), hydroxyl terminated polyether (HTPE), and polybutadiene-acrylic acid-acrylonitrile terpolymer (PBAN) have been conducted. While no definitive answer was found, the class of propellants based on ammonium perchlorate AP as oxidiser and HTPB has become the most widely used thanks to its high performance, ease of use, production and moulding, and its good mechanical properties.

AP can be used as a single ingredient for a monopropellant, however its final combustion products are composed of 30 % of oxygen, which could be used to burn extra fuel and increase the heat release. At high temperatures, the binder usually decomposes into several hydrocarbon species which can react with this excess oxygen. Hence, using the binder as an additional fuel takes advantage of this oxygen content to release additional energy via the combustion of the binder. This results in an increase of the combustion temperature.

Let us shortly focus on the family of AP-HTPB compositions. AP is used in the form of pellets, whose diameters are usually distributed in two or more classes: 150-200  $\mu\text{m}$ , and  $< 50 \mu\text{m}$ . Thus, small pellets fill the gaps between the larger ones, allowing for an efficient packing within the available volume. The inert binder then fills the remaining gaps. The flame structure of a composite propellant is complex, because it is inherently three-dimensional. Figure 1.10 shows the flame structure of an AP-HTPB composition. The AP pellet at the center undergoes a pyrolysis reaction around 600 to 720 K, which transforms the solid phase into ammonia ( $NH_3$ ) and perchloric acid ( $HClO_4$ ). These species react in a premixed manner directly above the pellet surface within a flame which is only a few micrometers thick, also referred to as AP monopropellant flame. The resulting combustion products are mainly  $NH_3$ ,  $HCl$  and  $O_2$  ( $\approx 30\%$ ). At the interface between the binder and an AP particle, reactions between AP and HTPB decomposition products form a first diffusion flame. The oxygen that remains after the AP monopropellant flame will react in a final diffusion flame with the decomposition products of the binder. The final combustion products are mainly  $CO$ ,  $CO_2$ ,  $H_2O$ ,  $H_2$ ,  $HCl$  and  $N_2$  and the overall flame height is on the order of 500  $\mu\text{m}$  at 20 bars. Since the final flame is of diffusive nature, this height tends to increase with pressure because of the blowing effect from gases coming from the AP premixed flame [17].

These combustion products still have the potential to react with other species. Hence it has become common practice to add small metal particles to the base mixture. These particles melt near the surface and are convected through the various flames by the surrounding gas flow. They typically travel along a length of 1 cm during their burning with the aforementioned products [14], further increasing the energy release and final temperature. Aluminium is very widely used with AP-HTPB propellants. Its combustion leads to the formation of alumina particles  $Al_2O_3$ .

The improvements obtained through these various additions can be seen by considering the final temperature of the combustion products. Pure AP combustion yields a flame temperature close



**Figure 1.10** Structure of the flame for an AP-HTPB composite propellant [6]

to 1500 K. The addition of the inert binder increases it to roughly 2500 K, and the addition of aluminium powder further increases it to 3500 K. As discussed previously, this higher temperature improves the motor performance.

High-energy composite solid propellants are obtained when a large part of the binder is replaced by highly energetic materials, such as CL-20 [1], and when some portion of the AP particles are replaced by HMX or RDX. These propellants provide an increase in specific impulse, but have mostly been used experimentally so far.

Finally, various additives, e.g. ferrocene  $Fe(C_5H_5)_2$ , may be included to adjust the burning rate and its sensitivities. More information is available in [6]. Note that composite propellants are usually impacted more strongly than homogeneous propellants by variations of the initial temperature of the propellant, i.e. the solid phase temperature deep below the burning surface.

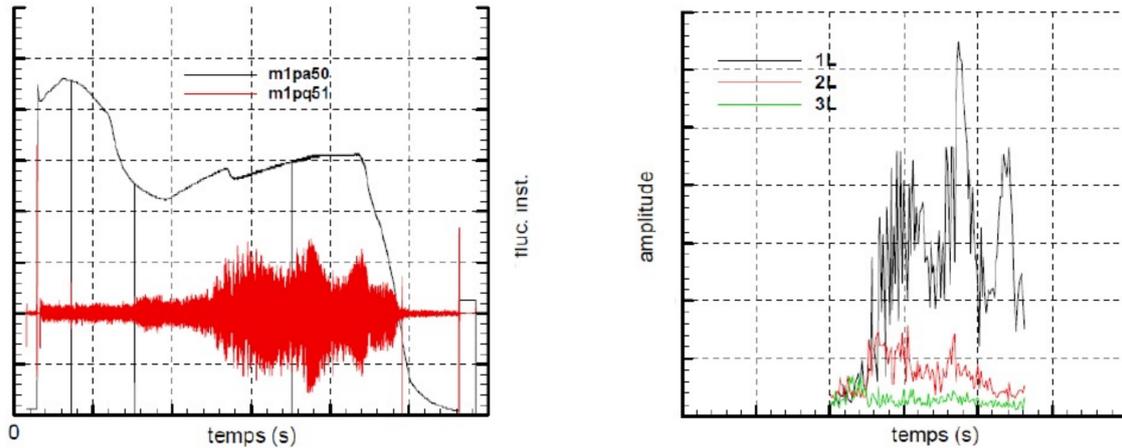
## 1.4 Main topics related to solid propulsion

The development of a large number of solid rocket motors since the 1950s, in particular the large segmented SRMs for the Space Shuttle and Ariane 5, as well as the variety of use cases and motor sizes have led to the discovery of multiple issues which may occur during a firing and that may be a cause of failure or bad performance. Consequently, large research efforts have been conducted and still are ongoing on each of these particular topics. We propose to list the main issues below, giving a short overview of the current understanding.

### 1.4.1 Unsteady dynamics and instabilities

Various test benches and SRM firings have shown that the internal flow field may be prone to unsteadiness and instabilities. This is apparent by fluctuations of the chamber pressure and consequently thrust oscillations. These instabilities may be harmful for the structural integrity of the

vehicle or its payload, especially in the case of small SRMs. The left side of Figure 1.11 shows the pressure history during one of Ariane 5 firings. Oscillations of about 1% of the chamber pressure can be observed. The right side of the figure shows the projection of the pressure fluctuations on the first three longitudinal acoustic modes of the chamber. Ariane 5 has a special device composed of a large number of resonators, which allows for the vibrations to be partially absorbed before they reach the payload, however this device adds inert mass to overall launcher weight, as well as more complexity. Therefore, the reduction of combustion chamber instabilities is still a subject of intense research.



**Figure 1.11** Left: Pressure signal (black) and fluctuating part (red) [89]. Right: decomposition of the fluctuations on the first 3 longitudinal modes.

The instabilities can be of various natures, which are shortly listed in the following.

### Acoustic instabilities

In this class of instabilities, the pressure fluctuates spatially and temporally. For an instability of wavelength  $\lambda$  in a chamber of length  $L$ , if  $\lambda/L \gg 1$ , the associated mode is called the  $L^*$ -mode or Helmholtz mode. It corresponds to fluctuations that are spatially uniform with a low temporal frequency. Those can typically be captured by the volume-filling approach presented in Section 1.2.3.

If  $\lambda/L \approx 1$ , the instability corresponds to a so-called acoustic mode, either longitudinal, radial or transverse (typical of smaller motors). Larger motors are essentially prone to instabilities in the longitudinal modes, as shown in Figure 1.11. For typical SRMs, good approximations of the modes are obtained by solving the Helmholtz equation in an equivalently-sized cylindrical chamber [90]. Analysis of the interaction between these acoustic modes, the internal flow field and the heat release via combustion allows to detect potential instabilities for a given geometry and propellant formulation.

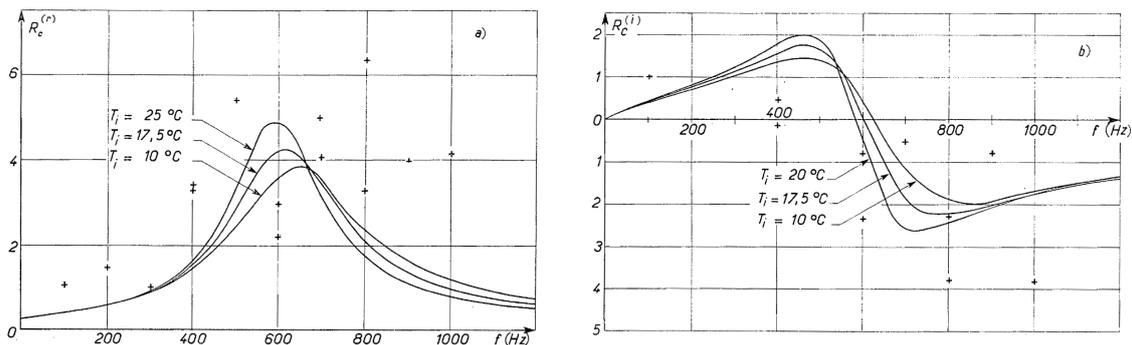
### Solid propellant combustion instabilities

When an acoustic instability is present, pressure fluctuations at the burning surface will cause fluctuations of the heat and mass release at the surface. In the case of low temporal frequencies for the acoustic instability, the quasi-steady burning behaviour from Equation (1.5) shows that the injected mass flow rate coming from the propellant consumption increases with pressure ( $n > 0$ ).

At higher frequencies, the mass flow rate increase may be even larger due to unsteady effects of the propellant combustion, and an important phase delay may appear. These effects can be represented by the pressure response function of the propellant for a pressure oscillation at the frequency  $f$ :

$$R_{mp}(f) = \frac{m'_s/\overline{m}_s}{p'/\overline{p}} \quad (1.17)$$

where the mass flow rate and pressure are decomposed in a fluctuating and a mean part, e.g  $p(t) = p' \exp(2i\pi ft) + \overline{p}$ . A typical response function, deduced from linear theoretical analysis, is plotted in Figure 1.12.



**Figure 1.12** Typical response function  $R_{mp}$  for a solid propellant (left: real part, right: imaginary part) [91]

For small or laboratory scale motors, the response of the propellant combustion may have an effect on pressure fluctuations [92]. For larger motors, the coupling between acoustic modes and combustion response is usually small [93] and the observed instabilities are thus mainly due to a coupling between the chamber hydrodynamics and acoustics. Still, research remains active on this topic [94, 95]. The modelling and the numerical tools presented in this work can be used for the study of the effect of propellant response on motor stability, see for example Section 6.1.

### Interactions between the internal aerodynamics and acoustics

Firings from the Titan34D launcher and of the Space Shuttle have lead to the observation of instabilities, which have directed the research towards aerodynamically-driven instabilities where vortices in the internal flow field interact with the chamber acoustics [96]. These have been experimentally assessed [97, 98] and numerically investigated, partially at ONERA [92, 99]. Research has lead to the conclusion that these instabilities arise from a so-called vortex shedding, i.e. vortices that are generated at the propellant surface along the length of the chamber. As they detach and interact with the nozzle, they may couple with acoustic perturbations, potentially leading to strong pressure oscillations. Vortex shedding has been classified in three categories:

- **Angle Vortex Shedding:** in that case, vortices are generated when the internal geometry has an abrupt variation in direction, creating a shear layer, as typically encountered at the end of a propellant load, before the nozzle. The impact of these vortices on the nozzle will create pressure waves in the chamber [100]. It has been experimentally and numerically investigated [9, 12], and was found to be the cause of instabilities in some military motors.

- **Obstacle Vortex Shedding:** here, vortices are generated when the flow encounters protrusions, such as layers that separate the segmented parts of the solid propellant load in a large SRM. These layers are not eroded as quickly as the surrounding propellant and therefore protrude in the internal flow field, emitting vortices at a certain frequency. Experimental and numerical investigations have however mitigated the importance of that phenomenon in segmented motors [92, 101].
- **Parietal Vortex Shedding:** in this case, vortices are generated by a gradual destabilisation of the internal flow above the propellant surface. The flow of gases injected perpendicularly to the surface quickly reorients along the longitudinal direction. When the chamber is sufficiently long, this reorientation gives rise to the spontaneous emission of vortices that depart from the near-surface region and develop inside the chamber. This instability is explained by purely hydrodynamic considerations, and was theoretically investigated in the work of Casalis et al. [10, 102]. For a chamber of length  $L$  and radius  $R$ , this phenomenon was found to occur if  $L > 8R$  approximately.

These vortex shedding mechanisms create vortices that can excite the acoustic modes of a combustion chamber by generating pressure waves as they impact the nozzle [103]. These pressure waves can then favour the appearance of additional vortices, leading to a sustained interaction between the pressure oscillations and the vortex shedding. As the internal geometry and the flow field evolve during a complete firing, there are specific moments where this coupling may be particularly important [10]. It has however been shown that the development of turbulence through the flow field has a stabilising action by helping dissipate the vortices [104], in particular for motors with a large aspect ratio  $L/R$ . This has encouraged the use of smaller motors for Vega and Ariane 6, which are less prone to vortex shedding.

### 1.4.2 Fluid-structure interactions

Pressure fluctuations, due to the aforementioned instabilities, or arising from the ignition transient, lead to a mechanical loading of the propellant grain and the motor casing. For instance, a feedback loop between internal pressure oscillations and structural deformations has been shown to exist on the Titan IV heavy launcher [105]. The failure of Titan IV PQM-1 test motor in 1991 [106] was successfully investigated using a simulation framework that couples the resolution of the fluid flow and that of the structural deformations [107]. Several studies have focused on the deformation of the propellant casing during ignition [56, 58], where the pressure fluctuations are particularly important. The pressure fluctuations may also cause unsteady motions of the protruding separations in the case of a segmented motor, which may increase the emission of vortices via vortex shedding and thereby increase the overall pressure fluctuations [108]. Finally, dynamic deformations of the propellant can couple with the propellant flame dynamics and impact the response to pressure oscillations [94].

This topic is still an active area of research. The numerical study of fluid-structure interactions requires the coupling of a mechanical solver, for the solid propellant and motor casing, and a fluid solver for the internal flow field. In the present work however, deformations of the solid propellant grain and motor will not be considered.

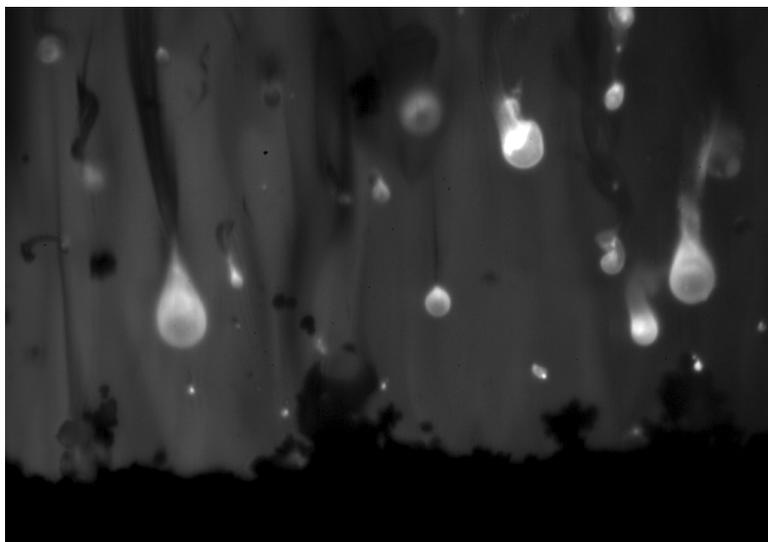
### 1.4.3 Multiphase distributed combustion of aluminium particles

In metallised propellants, in particular aluminised propellants such as the AP-HTPB-Al mixtures found in most large SRMs, metal particles do not burn in the propellant flame itself.

During manufacturing and storage, these particles oxidise, forming an impermeable nanometric layer of alumina  $Al_2O_3$  at their surface. During a firing, once the burning propellant surface has regressed and uncovered an aluminium particle, this particle is heated up until it reaches its melting point at 933 K. In the course of this temperature rise, thermal dilatation causes the alumina layer to crack, exposing unoxidised aluminium. While at the surface, several melted particles may agglomerate into a larger one.

At one point, the particle detaches from the surface and is convected by the surrounding gas flow. It starts to oxidise with the gaseous species from the latter, producing submicronic alumina smoke particles, as well as gaseous oxides such as  $AlO$ ,  $Al_2O_2$ ,  $Al_2O$ ,  $AlO_2$ ,  $CO$ . Oxidation may also occur when the particle moves through the AP premixed flame where oxygen is present in large quantities.

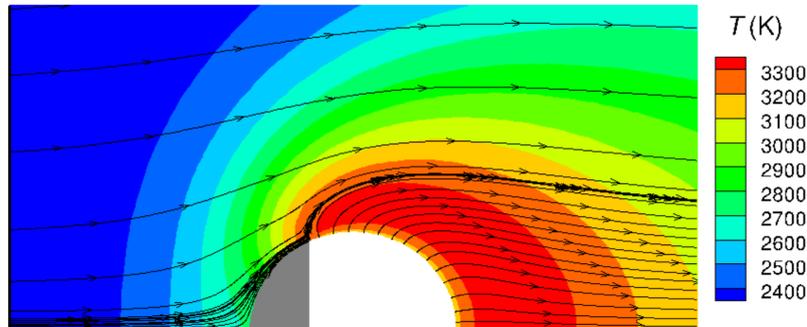
The oxidation process is highly exothermic and therefore greatly increases the combustion temperature. As the particle reaches the melting point of alumina at around 2300 K, the alumina liquid phase, which is not miscible with liquid aluminium, forms a lobe at the surface of the droplet, greatly reducing its isolating effect. Ignition follows, and oxidation continues in the atmosphere of propellant combustion products, mainly with  $CO_2$  and  $H_2O$  [109], forming a diffusion flame around the droplet. The particle shrinks as it burns. Combustion ends when only liquid alumina remains, which typically corresponds to a particle diameter of a few micrometers. Note that agglomerated aluminium particles may form larger residues, on the order of 50  $\mu m$  [110]. All the previous phenomena can be observed in Figure 1.13, which shows aluminium particles leaving the propellant surface. A large body of experimental observations have led to the determination of combustion laws to approximate the lifetime of a burning solid particle. A law such that the rate of consumption of aluminium for a particle of diameter  $d$  is proportional to  $d^n$ , with  $n \approx 1.8$ , reproduces these experiments with good accuracy [111, 112].



**Figure 1.13** Ombroscopy visualisation of aluminium droplets burning above the propellant surface (©ONERA, courtesy of Robin Devilliers)

The reaction mechanism for the diffusion flame and the heterogeneous surface reactions that may occur are not well known, in particular when considering a surrounding atmosphere composed of solid propellant combustion products at high pressures. Recent research has been directed towards a better understanding of these phenomena [112], with specific attention to the evolution of the

alumina lobe, as its shape may affect heat and mass transfers in the diffusion flame. An example 2D simulation of a droplet combustion is shown in Figure 1.14. ONERA and ArianeGroup continue this research with a large body of experiments to gather data on aluminium droplet combustion and agglomeration, as well as numerous computational studies [113, 114].



**Figure 1.14** Simulated temperature field around a burning aluminium particle of diameter 120  $\mu\text{m}$  with an alumina lobe (in grey) [112]

Other research focus is on the effect of the droplets on the internal flow, with a strong numerical effort by ONERA and collaborators [14, 15, 115]. For instance, residue particles may impact the flow field by damping or amplifying acoustic instabilities [13]. In such work, aluminium particles are represented as a dispersed phase, and an Eulerian resolution of the William-Boltzmann equation is used to produce a statistically representative behaviour of that phase, accounting for evaporation/-combustion and coalescence of the droplets [14]. The combustion of aluminium droplets may also give rise to thermoacoustic instabilities [11, 116, 117]. Finally, droplets may accumulate at the rear end of the chamber, near the nozzle [118], impacting the motor performance.

#### 1.4.4 Numerically-driven propellant formulation

Up to now, the propellant compositions have been formulated using an empirical approach, based on trial and error. Existing combustion models are not yet predictive enough, and are mostly used for studying the effect of minor parameter changes around a known configuration.

There has been a large body of work in the US to create propellant combustion models that are generic and may be used to study the combustion of any combination of a few basic ingredients, such as RDX, HMX, AP, HTPB. Such an attempt at a generic model is the BYU universal mechanism [17, 33, 37] which contains 106 species and 611 reactions. Good agreement with experimental data has been demonstrated, however the condensed degradation mechanism is still insufficiently known.

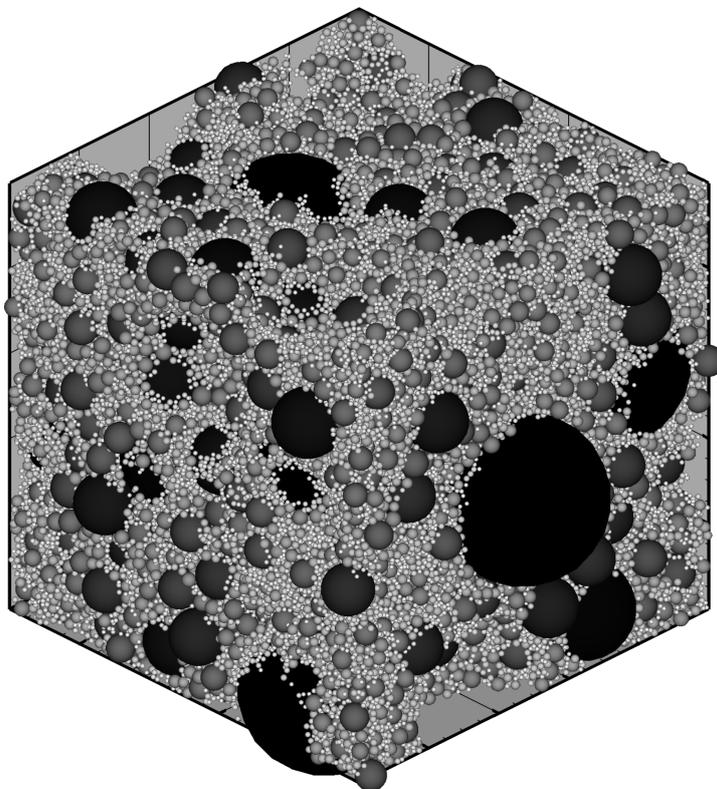
Most work has focused on homogeneous propellants, for which a 1D approach is well suited. However, due to the complex reaction mechanisms and diffusion processes, such a detailed model makes it computationally impossible to envision 3D simulations for heterogeneous propellants. Most reports using detailed gas phase kinetics for heterogeneous propellants rely on a 1D approach [33], thus effectively considering only premixed flames whereas, in reality, diffusion flames are present (recall Figure 1.10). Thus, this premixing may lead to erroneous results at high-pressure, where it has been show [17] that the final diffusion flame is pushed back and does not impact the propellant decomposition any more.

Only a few studies have focused on 2D configurations but are limited to steady-state combustion with a flat propellant surface [17, 119]. For heterogeneous propellants, the effect of the non-uniform

surface regression and of the spatial distribution of the different ingredients has a major impact on the propellant combustion characteristics, thus a 3D approach is required to be able to predict the effect of different AP particle sizes for instance, as discussed next.

### 1.4.5 Numerical studies of heterogeneous propellants

For composite propellants, there has also been a considerable amount of development to build numerical tools able to accurately represent the burning at macroscale, taking into account the shape of the various particles and their size distribution. Models have been mostly presented in the US with the code ROCFIRE [19, 120, 121] and in France with the codes COSMETIC at ArianeGroup [39, 122, 123] and COMPAS at ONERA [20, 124]. To obtain a statistically representative spatial distribution of particles and binder, random packing algorithms are used, producing particles packs similar to that of Figure 1.15.



**Figure 1.15** Pack of  $10^5$  AP particles [124]

Much work has been done with these tools. Having three-dimensional simulation tools enables a DNS approach (direct numerical simulation) for the study of composite propellant combustion. Much information can be gained from simulation results, which can help feed simpler models, or understand the effect of microscale parameters. For instance, velocity fluctuations in space and time above the propellant flame have been studied [123, 125, 126], with the aim of defining proper boundary conditions for LES simulations of a whole combustion chamber. Such DNS simulations have shown that the spatial fluctuations happen on length scales (1 mm) which are negligible compared to those of the combustion chamber and its flow field. However, temporal point-wise fluctuations may be significant, and occur on time scales (1 ms) comparable to time scales of the flow. For

large-scale motor simulations, where a detailed representation of the propellant surface is computationally impossible, a boundary model can be developed, which produces a so-called combustion noise, reflecting the temporal fluctuations observed near the propellant surface in detailed simulations. Prior to the previously cited works, constant injection conditions with a potential addition of white noise were used instead. It has been shown that the newly obtained temporal fluctuations may affect the acoustics of a motor at low frequencies [127]. It has been also shown that these fluctuations could help trigger instabilities [123].

To construct a one-dimensional model of an actually heterogeneous propellant, it may be required to compute effective thermal properties based on the knowledge of the properties of each separate component (AP and HTPB for instance). This can be done by simulating the heat transfer in a sufficiently large spatial representation of the heterogeneous structure, and fitting effective properties to the results [128]. Other work [129] has focused on the validation of the homogenisation approaches used in widely spread analytical burning rate models (e.g. BDP model [130]). In [131], detailed 2D microscale simulations are conducted and space-averaged to feed a one-dimensional model, so that the latter retains surface mass flow rate and heat flux values that are representative of the ones obtained by spatially averaging multidimensional propellant combustion simulations.

In the articles mentioned in this subsection, all computations are carried out with simplified gas-phase kinetics, inspired by the BDP model [130]. These global kinetics can however be made reasonably accurate by properly fitting their parameters to the results obtained previously mentioned 2D simulations with detailed kinetics. This has for example been presented in [132] for a 6-species and 4-reactions chemical mechanism, with comparison to detailed 2D simulations [133] which consider 37 species and 127 reactions.

The above simulations mainly focus on AP-HTPB combustion and assume that AP particles are spherical, which may not be accurate, in particular for the smaller particles. In [134], the effect of a change in particle shape and orientation was studied, and it was found to be relatively negligible, as long as the overall packing of particles is sufficiently randomised.

Overall, detailed microscale simulations are useful to construct simplified models that retain properties or dynamics reminiscent of those observed in the DNS simulations. A detailed review of the investigations conducted with such simulations is given in [135, 136]. Still, these models and tools are highly complex and are constantly evolving. A larger body of work in all areas, e.g. kinetics, modelling, numerical methods, and experimental observations, is required to envision the development of truly predictive tools [21, 137].

### 1.4.6 Transient phenomena

The term transient denotes the transition between two different states of the rocket motor. Strong pressure variations usually occur and may damage the structure of the motor.

The subject of the present manuscript, ignition, is one such transient. The highly complex and multiphysical ignition process is still difficult to model and predict. Simple analytical models [42] can only be fitted *a posteriori*, while more evolved simulation tools only recently appeared [54, 56] and require large amount of computational resources while still using relatively simple propellant models. Therefore, the design of the igniter system relies heavily on experimental results obtained with other motors, for example to obtain a correlation between the motor initial free-volume and the mass of the igniter charge required [1]. Recently, there has been a renewed interest in the modelling and simulation of this phenomenon, and the present work is part of this effort. This subject is dealt with in more details in Section 1.5.

Another transient phenomenon is the tail-off phase of an SRM, i.e. the time range where the thrust leaves the quasi-steady operating regime and the motor extinguishes. In that phase, specific academic configurations have shown to undergo strong vibrations. In the case of a space launcher, strong thrust variations could damage the upper stages and payload. Numerical investigations at ONERA and ArianeGroup are being conducted to study the impact of vortex shedding and propellant combustion response on those aspects. The propellant models and simulation tools developed in the present work could also be applied to such studies, as these tools are able to simulate the unsteady combustion of a solid propellant.

#### 1.4.7 Electromagnetic signature

The determination of the electromagnetic signature of a rocket plume, in particular for an SRM, has received a renewed interest in the past years. The combustion products that exit the chamber through the nozzle are rich in  $H_2$ ,  $CO$  and  $HCl$  which undergo post-combustion as they are released in the air surrounding the rocket. Alkali species such as  $K$  and  $Na$  contained in the propellant ingredients experience thermal ionisation, releasing electrons. The ionised rocket plume interacts with electromagnetic waves, potentially perturbing communications with the ground. Also, the plume exhibits an electromagnetic signature which is characteristic of the rocket. This signature can be used for radar detection. Research efforts are directed towards a better characterisation of the postcombustion flame, the electron density in the plume, the overall radio spectrum of the plume, both experimentally and numerically [16, 138]. On the simulation and modelling side, particular care must be taken for the turbulence and ionisation models, chemical mechanisms and spectral emission computations. Also, the presence of alumina particles must be modelled with appropriate multiphase considerations.

#### 1.4.8 Overview

Throughout the development of solid rocket motors, new issues have emerged. A large body of experimental, theoretical and numerical work is usually associated with each of these. While some issues have been very thoroughly investigated and can now be considered as decently mastered (e.g. vortex shedding), others continue to present a challenge, either because of the wide variety of phenomena involved, or because of the limitations of the experimental observations, or a combination of both.

Still, the extensive progress achieved through the various research programmes has had a clear impact on reliability, as demonstrated by the high reproducibility of solid rocket motor firings and their very low rate of failure [22], with a single space launcher failure in the last 20 years in the USA and Europe, as mentioned in the general introduction,

Ignition of the solid propellant load is, however, still a challenging problem, in particular due to the complex multiphysics phenomena involved. The ignition transient is usually very short (less than 1 second) compared to the complete combustion duration of the main propellant load (on the order of one minute). However, the simultaneous occurrence of highly unsteady phenomena such as igniter gas flow, surface heating, pressure waves, ignition of the main load and subsequent flame spreading makes this ignition phase critical for the overall vehicle. Indeed, the strong pressure variations, in particular the rapid pressurisation, lead to high unsteady mechanical and thermal loads on the propellant and motor structure, which can lead to the failure of the rocket, exemplified by the Titan IV PQM-1 test motor in 1991 [106]. Also, there is a risk of misfire, which can lead to

an unstoppable slow burning of the complete load.

The development of a new motor still involves a costly trial and error process, and ignition is still problematic in some configurations. It is therefore essential to have numerical tools that are able to correctly predict the ignition transient, so as to identify potential issues early on. In the next section, we take a more in-depth look at this particular phase.

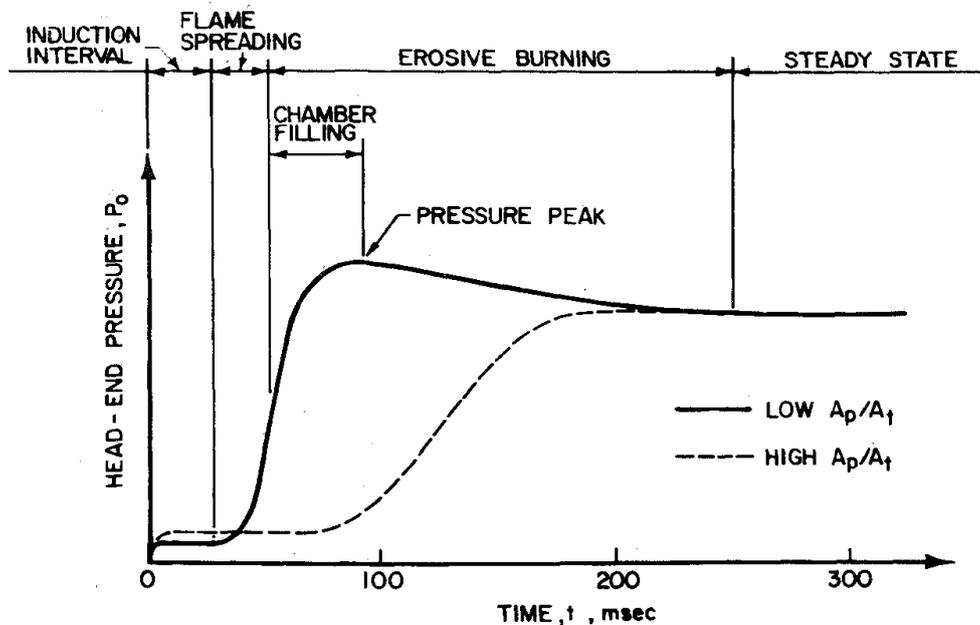
## 1.5 Ignition of SRMs

The ignition of a solid rocket motor generally refers to the time interval between the start of the ignition sequence and the attainment of design operating conditions. It involves multiple systems (igniter, nozzle cap...) as well as physical and geometrical aspects (physico-chemical properties of the propellant, grain geometry). The correct prediction of the ignition of an SRM also has a strong prerequisite, which is the prediction of the ignition of the propellant itself, without the interaction with the combustion chamber flow field. That latter part is conveniently studied through small laboratory experiments.

In this section, we shall first discuss the main aspects of ignition in an SRM, before addressing in a second part the ignition of a solid propellant sample. Then, we conduct a thorough review of simulations of SRM ignition presented in the literature and, eventually, the position of the present work is discussed accordingly.

### 1.5.1 Phenomenology of the ignition transient of an SRM

The ignition can be subdivided into various intervals as follows: induction time, flame spreading, chamber filling, stabilisation on steady-state, as shown in Figure 1.16.



**Figure 1.16** Typical pressure history during ignition for large or low port-to-throat area ratio motors[46]

**Induction** This interval corresponds to the delay between the start of the igniter system and the first ignition of the main load. The ignition is initiated by the *igniter* system, of which there are several kinds. The first one is pyrotechnic igniters. These generally consists of a perforated casing holding high-energy charges in the form of pellets, typically consisting of black powder with a binder. These pellets are ignited electrically, and their combustion products (hot gases and particles) eject through the perforations of the casing, towards the main propellant load in the combustion chamber. This load is heated by convective<sup>1</sup> heat transfer from the flow of hot gases, radiation from the hot particles and conduction upon impact of these particles on the propellant surface. Another class of igniters is the pyrogen one. Such igniters are actually small solid rocket motors used to generate hot gases which are expelled at high speed towards the surface of the main load. The dominant mechanism of heat transfer is therefore convection. This type of igniter will be the only one considered in this work. The igniter is usually placed at the head end of the motor (see Figure 1.3), however there are configurations where it is located at the aft end, which may result in a somewhat different ignition sequence [139]. In the present work, the focus will be on head-end pyrogen igniters, however the simulation tools developed are equally applicable to aft-end pyrogen igniters.

The main propellant load, initially at a low temperature (typically 300 K) is heated up by the various heat fluxes (convective, conductive, radiative). As a given point at the surface reaches a certain temperature level (dependent on the physico-chemical properties of the propellant and on the operating conditions), the decomposition of the propellant (pyrolysis) becomes sufficiently important and pyrolysis products are injected in the gas phase above the propellant. These products react together and form a flame which attaches to the propellant. The associated heat flux is larger than the previous convective, radiative and conductive fluxes, and becomes the dominating surface heat flux for the ignited portion of the propellant.

**Flame spreading** Upon ignition of a point at the propellant surface, the propellant flame creates additional heat fluxes (convective and radiative) that increase the propellant heating rate in the nearby unignited region. Thus, rapid ignition of the surroundings occurs, which then spreads to the whole motor. The flame spreading interval corresponds to the delay between the first ignition and the ignition of the whole propellant surface.

**Chamber filling** This period corresponds to the time between the end of flame spreading and the attainment of the steady-state design operating pressure. Note that there might be situations where the induction, flame spreading and chamber filling actually overlap, especially if some regions of the propellant ignite more slowly.

Due to the accumulation of igniter gases and combustion products from the ignited portions of the main load, the chamber pressure increases rapidly, as can be described with the simple mass balance (1.10). Therefore, the rising pressure  $P_c$  induces an increase of the burn rate as dictated by the Vieille law (1.5), which in turn increases the pressure rising rate. This positive feedback loop is referred to as *dynamic burning*. If the ignition has not yet spread to the whole grain, the propellant mass flux term can, at first-order, also be affected by the burning area  $A_b$  that increases in time, further enhancing the dynamic burning effect.

Finally, especially in motors with a large length-to-diameter ratio, high cross-flow velocities may be found above the propellant surface, typically near the aft-end of the motor. This cross-flow may

---

<sup>1</sup>“Convective” indicates that the fluid and the surface that exchange heat are in motion relative to each other, however the actual physical heat transfer mechanism remains conduction.

locally enhance the burning rate of the propellant, an effect which is referred to as *erosive burning*, and traditionally attributed to a turbulent enhancement of the diffusion processes near the surface, thus [140, 141].

The sum of all these effects may result in a temporary pressure overshoot, as depicted in Figure 1.16. Most notably, the maximum pressure reached during that overshoot must be precisely known as the mechanical structure must be designed to withstand it.

Some motors also possess a nozzle cap, which initially blocks the nozzle and breaks apart at a certain pressure level. This initial closure of the combustion chamber increases the pressurisation rate. The nozzle seal may also increase the importance of pressure waves produced by the initial igniter gas flow, as these waves will not partially escape through the nozzle but will be reflected inwards.

**Stabilisation at steady-state** Once the ignition process has completed, the potential excess pressure will be evacuated and the chamber conditions will settle on a slow quasi-steady evolution, where the pressure mainly depends on the evolution of the propellant burning area  $A_b$ , similarly to the equilibrium pressure derived in Equation (1.11).

## 1.5.2 Solid propellant ignition

The ignition sequence described in the previous section depends on various parameters, which are mainly the chamber geometry, the igniter system and the propellant load ignition characteristics.

A good description of the latter is required if a decent prediction of the ignition transient of an SRM is to be obtained. Indeed, various propellant formulations behave differently in terms of initial heating rate, time to first ignition, flame appearance and flame propagation. The following paragraphs thus provide an overview of the physical phenomena involved in ignition.

### Physico-chemical mechanisms of ignition

When a solid propellant sample, initially at ambient temperature  $T_0$ , is heated up at its surface by an external stimulus (laser, hot flow impingement, contact with a hot solid), the surface temperature initially rises like that of an inert material. For instance, in the case of constant laser heat flux absorbed at the surface, and for constant thermal properties, the surface temperature  $T_s$  is of the form:

$$(T_s(t) - T_0) \propto \sqrt{t} \quad (1.18)$$

When the temperature at the surface becomes large enough, chemical reactions start to occur and pyrolysis gases may be expelled from the surface. Three classes of reactions are usually identified:

- **in-depth condensed phase reactions**, where heat is released in the solid phase (also referred to as condensed phase) via exothermic reactions ;
- **surface reactions**, where heat is released at the surface via reactions with the expelled pyrolysis gases or ambient oxidiser surrounding the sample ;
- **gas-phase reactions**, where a gas flame is formed as the pyrolysis products react together, providing a strong heat feedback to the propellant surface.

A propellant may be ignited via thermal runaway caused by any combination of the previous heat release sources. Depending on its composition, one of these sources may be predominant in achieving ignition. This may also depend on the external stimulus. For instance if the sample is exposed to a strong cross-flow of inert gases, the gaseous pyrolysis products may be blown away from the surface before the flame can be established.

In the case of predominant solid-phase reactions, as initially studied by Frazer [142], ignition occurs when the heat released by these reactions is larger than the heat lost to the deeper layers of the propellant and to the surroundings, thus yielding an ignition temperature and delay which depend on the external stimulus, but are independent of the gas-phase pressure, since solid phase reactions are pressure-independent.

Better reproduction of experimental ignition times for certain propellant compositions was achieved by considering heterogeneous surface reactions between the propellant and the surrounding gases [143], supported by the experimental observation that samples surrounded by an oxidising atmosphere ignite more quickly as the oxidiser concentration increases. This model helped with the analysis of hypergolic ignition, i.e. ignition of a solid propellant by reaction with an oxidising atmosphere, a technology which was studied in the 1960s. Note that the vast majority of solid rocket motors are initially pressurised with inert gases such as nitrogen, hence this approach is not relevant for these.

Finally, ignition may also be controlled by the appearance of a gaseous flame above the surface. In that case, the surface initially expels pyrolysis products, at a rate dependent on the surface temperature. As the surface is heated up by an external heat source (e.g. laser, convective flow), the gradual emission of pyrolysis products leads to the formation of a gaseous mixture which spontaneously ignites after an induction period. The flame that is generated then attaches to the surface, producing a strong conductive heat transfer to the propellant. The heating rate then momentarily greatly increases, until the pyrolysis rate becomes sufficiently strong and pushes back the flame, leading to a stabilisation of the surface heat transfer. This mechanism was first studied theoretically in 1965 [144].

## Modelling

Historically, various theoretical ignition models have been established based on each of these three approaches and using a one-dimensional representation of the solid and gas phases, as reviewed in [4, 18]. However no model has yielded good results for all propellant types, and they all were semi-empirical in nature, requiring extensive experimental data for their parameters to be properly adjusted [18].

In the 1990s, various detailed numerical ignition models have emerged for homogeneous propellants, utilising complex kinetics for the gas-phase and surface reactions, and simpler mechanisms for the solid phase [18, 32, 145], taking as framework the latest steady-state propellant combustion models, e.g. [146, 147]. The review [18] gives a detailed overall picture of the various ignition models, from the early analytical models to later detailed models. Extensions to composite propellants have also been successful [33]. This modelling will be presented in greater details in Chapter 2.

All these models assume a one-dimensional space representation. Two- and three-dimensional models have emerged for composite propellants in the 2000s, in particular thanks to the important work by Massa and Jackson [19, 120, 148], with simplified global kinetics. A single application to laser-induced ignition has been reported by ArianeGroup [39] for AP-HTPB propellants, showing that ignition is driven by the appearance of the AP/binder primary diffusion flame.

All this modelling work is focused on reproducing the ignition of small propellant samples. The models are usually highly complex, either in terms of kinetics, phenomena considered (e.g. multiphase flow), or geometrical considerations (multidimensional domain for composite propellants), and they are usually associated with a large computational cost [17, 34, 149]. As we will see in the next subsection, this has led to the use of simpler ignition models for simulations of complete SRMs.

### 1.5.3 Simulating the ignition transient of an SRM

Numerous issues have motivated the development of simulation tools to study the ignition of SRMs. Among these issues, we can mention the failure of test motors [106], misfires, or unexpected overpressures experienced at the launchpad during ignition [60].

The need to simulate the ignition of SRMs has motivated a large body of research since the 1960s. Due to the lack of computational resources, numerical methods, modelling and experimental knowledge, early models were very simplified, assuming a 0D chamber. As knowledge and computational capabilities progressed, more refined models appeared, with one-dimensional combustion chambers. Finally, truly multidimensional simulations appeared in the 1990s. In this section, we will review the main models that have been presented in the open literature, and discuss the position of the work presented in this manuscript.

#### Early zero-dimensional models

The early research has focused on zero-dimensional models of the combustion chamber [42–45], where the evolution of the chamber pressure is governed by an ODE, constructed from a simple mass flux balance between the burning propellant surface and the nozzle exhaust, similarly to what has been presented in Section 1.2.3. Thus, they are referred to as volume-filling approaches. Such studies give a first insight into the dynamics of the chamber pressure during ignition, but do not accurately capture the flame spreading over the propellant surface.

Some simplified analytical models [42] were presented to account for the longitudinal propagation of ignition. A more complex model is presented in [43], accounting for a uniform gas-phase pressure and piecewise linear longitudinal flow velocity in the combustion chamber, with a non-uniform solid propellant domain where heat conduction occurs both radially and longitudinally. This allows to compute the ignition transient and the flame spreading on the propellant surface, by linking the surface convective heat flux to the chamber pressure and local flow velocity, via the use of ad-hoc heat-transfer coefficient correlations, inspired from the formulations obtained for turbulent pipe flows. The propellant is considered locally ignited upon reaching a specified ignition temperature, and is subsequently set to steady-state burning following a Vieille law. The results, though limited, have shown that longitudinal heat conduction within the solid propellant has a very negligible effect on the ignition transient. In a subsequent study [45], the average chamber gas temperature is allowed to vary in time, and the flame spreading is simplified by considering a constant flame-spreading velocity, which must be iterated upon to match experimental measurements.

Overall, these early models give useful insights into the ignition dynamics of SRMs, but are too simplified and require extensive data to be fitted. No truly predictive simulation is therefore possible with this approach, in particular for motors with large length-to-diameter ratios, where a high velocity flow and pressure non-uniformities are observed experimentally.

## One-dimensional models

To improve the reproduction of flame spreading and limit the number of ad hoc formulae required, a change of modelling paradigm was required. Also, strong effects from pressure waves induced by the igniter start-up could not be captured by the zero-dimensional approach. Following the advancement of computational capabilities and numerical methods, such as the emergence of numerical schemes for the solution of the unsteady Euler equations, transition to one-dimensional chamber models has been possible, where physical fields are allowed to vary along the length of the motor.

For instance in [46, 47], the unsteady one-dimensional flow in the combustion chamber is coupled with the unsteady heating of the propellant along the length of the chamber. Following the previous observation that lateral heat conduction within the solid phase is negligible, the heating of the propellant can be represented in 1D only, in the radial direction. To account for the non-uniform propellant surface temperature in the longitudinal direction, one such 1D model is used at each surface point of the mesh used to discretised the combustion chamber. This locally one-dimensional representation of the propellant thermal profile has become standard and, as we will see further, is the only approach used nowadays. In the work of Peretz [46], to avoid discretising the in-depth propellant thermal profile at each surface point, the propellant temperature profile is assumed cubic with respect to the depth below the surface, and a simple ODE on the surface temperature at each surface point is obtained. Heat transfer from the gas phase to the propellant is governed by empirical correlations obtained for turbulent flow in pipes. Upon reaching a surface temperature of 700 K, instantaneous ignition of the propellant is assumed, and its regression rate is assumed to be that of steady-state (Vieille law). Erosive burning is taken into account by augmenting the standard Vieille law with a term involving the local flow velocity. For instance, Peretz [46] uses the following corrected formula, inspired by theoretical work from Lenoir and Robillard [150] on the heat transfer experienced by a transpiring surface exposed to a longitudinal cross-flow.

$$m = aP^n + kh \exp\left(-\beta \frac{m}{(\rho u)_{longi}}\right) \quad (1.19)$$

with  $h$  the heat-transfer coefficient,  $k$  is evaluated based on theoretical friction analysis, and  $(\rho u)_{longi}$  is the gas-phase mass flow rate parallel to the propellant surface. The coefficient  $\beta$  is adjusted manually to fit experimental measurements. Note that the prediction of erosive burning is not a prime objective of the present work, hence this problematic will not be considered further, even though a few possibilities to model that effect will be briefly discussed at the end of this manuscript. The results presented [46] for a small experimental motor with a large aspect ratio show that the simulation agrees well with the measurements if the erosive burning contribution is included.

An overall review of the main early ignition models can be found in [139]. All these early models require extensive experimental data to be properly fitted, and cannot reproduce accurately the various phenomena or account for complex grain geometries. Also, such transient 1D chamber models require multiple modelling choices, in particular with regard to the igniter flow impingement on the propellant surface, which is often computed by assuming a specific shape for the impingement region, deduced from geometrical considerations [48]. Furthermore, the surface heat transfer is modelled via steady-state heat exchange laws with coefficients obtained by empirical correlations. These require fine-tuning and cannot accurately reproduce the unsteady heat transfer at all positions along the propellant surface for all flow conditions. Finally, actual propellant load shapes are often highly three-dimensional, for instance with the presence of aft-fin areas, as already discussed in Section 1.2.3. Simplified modelling of these zones is possible, for instance via 0D cavity models [48], however it introduces further parameters to tune and lacks good predictability. Other geometrical

peculiarities where the quasi-1D representation of the chamber flow is not valid (e.g. around the nozzle if the latter protrudes into the combustion chamber) require the introduction of submodels. Improvements have been made by coupling a one-dimensional flow field with a 2D tool to compute more geometrically complex zones, for example star-shaped regions of the grain as in [151]. In this work, good results are obtained for Titan IV and Shuttle SRM ignition transient. In particular, the model has enabled to reproduce for the first time the high initial pressurisation peak observed during the Shuttle SRM ignition, thanks to its ability to account for geometrical variations. This clearly demonstrates the importance of an accurate geometrical representation of the grain, if a correct prediction of the unsteady pressure level is to be obtained. However, the intricate modelling and solver coupling further complicate the simulation framework and still involve many correlation parameters.

Therefore fully two- or three-dimensional unsteady simulations of the ignition transient with accurate resolution of the near-surface phenomena (e.g. heat transfer) and geometrical peculiarities seem like the next major improvement in terms of reproduction fidelity, avoiding approximate and iterative modelling of the heat flux and flame spreading for instance. Nonetheless, evolutions of these 1D models have been investigated [48, 49, 152] and results may prove very good once the various coefficients have been adjusted. Their low computational cost makes them very appropriate for parametric studies around known configurations [49]. Fully predictive simulations can, however, not be envisioned with such tools.

## Multidimensional simulations

One of the first presentation in the open literature of such a simulation is given by Johnston in 1995 [50] for a 2D-axisymmetric configuration. In this work, the internal flow field is modelled by the Euler equations, discretised using with a finite volume scheme. The solid propellant is modelled with the one-dimensional heat equation at each boundary face, and this equation is solved using an integral formulation, requiring the storage of the entire time history of the surface heat flux. To compensate for the inviscid character of the flow, the convective heat flux from the gas to the solid is computed via an empirical correlation for turbulent pipes. Radiation from the gas phase to the surface is also taken into account to accelerate flame spreading, in particular for the ignition of the inter-segment cavities of the simulated Titan motors, where the convective heat flux is naturally low due to the locally slow flow speeds. Ignition occurs locally once the surface temperature of the propellant reaches a predefined ignition temperature. The propellant then switches to a steady-state Vieille law. After an iterative tuning of the radiation parameters, excellent agreement is obtained for the pressure history when reproducing test firings. However the star-shape of the simulated grain cannot be captured by the 2D-axisymmetric setup and is therefore modelled by modifying the burning rate dynamically, simulating a gradual ignition of the fins (linear time progression of the ignited fraction of each fin). The approach is however fully compatible with 3D domains, and the 2D setting was only retained for simplicity and computational efficiency.

At ONERA, 3D simulation tools with dynamic propellant ignition models have been developed during the 1990s. In particular, Bizot [61] presented the coupling of an ignition model inspired from previous work [27] with a 3D CFD solver. In that particular work, a subscale solid rocket motor is simulated on a 3D periodic mesh. The ignition model assumes that heat conduction inside the propellant is one-dimensional. The heat equation in the solid is discretised with a finite-difference approach, thus no approximation of the shape of the temperature profile (e.g. cubic with respect to space as in other works mentioned) is required, and the heat flux history need not be stored for an integral resolution. The propellant flame is a simple one-dimensional flame sheet model that enables

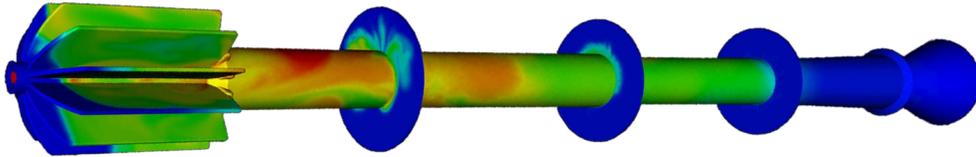
an analytical computation of the heat conducted by the gas flame into the propellant, depending on the surface temperature of the solid and on the wall pressure prescribed by the CFD solver. This flame allows ignition to occur dynamically, without resorting to a temperature criterion as in previous models. The ignition model can be fitted on its own to experimental laser ignition tests for instance, to correctly reproduce the dependence of the ignition time (or ignition delay) on the heat flux history. This reduces the number of parameters that need to be iterated upon to fit the overall 3D ignition result. Improvements to this approach have been presented in [51, 153]. The comparison to experimental results on subscale motor ignition still shows that the flame spreading occurs too slowly, and this is attributed to the lack of radiation modelling. Indeed, it is expected that the propellant flame, the burning aluminium particles and the aluminium oxide droplets will radiate heat towards the unignited zones, greatly increasing their heating rate.

An evolution of this framework is presented in [52] where radiation effects are included, by adding a radiative heat flux on unignited propellant boundary faces that are close to ignited ones. The 3D turbulent flow solver MSD (precursor of the current multi-physics CFD solver CEDRE from ONERA [71]) uses a k-L turbulence model. Wall laws are used to compute the heat transfer at the propellant surface, without resorting to the costly mesh refinement that would otherwise be required to properly resolve the turbulent boundary layer. The addition of the radiative heat flux allows to reproduce fast flame propagation speeds observed at the aft-end of an experimental motor that has a pressure cap initially blocking the flow at the nozzle throat. Without this effect, the convective heat flux is indeed too low in the second half of the motor because the nozzle closure slows down the flow in this region and does not permit the evacuation of the initial cold pressurisation gases (e.g. nitrogen). However, the dependence of the simple radiation model and flame spreading on the mesh refinement is not studied.

At the end of the 1990s and beginning of the 2000s, a large research effort has been conducted in the United States for the development of the fully-integrated multiphysics platform ROCSTAR, developed at the Center for the Simulation of Advanced Rockets (CSAR) for the simulation of ignition in SRMs. It couples the solid propellant deformation, heating and combustion, with the compressible multiphase flow in the chamber. In [53], first simulation results obtained with this new tool-set are presented. An ignition model is included, which consists of a 1D unsteady heat equation and an ignition criterion based on the surface temperature. The heat equation is solved approximately using a similarity assumption on the temperature profile inside the propellant, which is only accurate if the surface heat flux varies slowly with respect to the characteristic time of heat diffusion in the propellant. Simulation of the Space Shuttle SRM ignition without turbulence enhancement of the heat transfer yields a very slow ignition compared to experimental measurements. Therefore, the authors indicate that it is important to account for turbulence and properly resolve the turbulent boundary layers, which enhances the heat transfer to the propellant and leads to a faster ignition and higher pressurisation rates. Once the propellant ignites, the parietal injection of its combustion products blows away the boundary layer. Hence, from a numerical point of view, ignition simulations require a higher mesh resolution than simulations where the whole propellant is assumed already ignited at the beginning. Therefore, the authors present a first study of the turbulence-enhanced heat transfer on a simpler geometry, using the Baldwin-Lomax turbulence model. While the time of first ignition is unaffected by turbulence, the time to complete ignition is notably reduced with the addition of the turbulence model.

Further work using this code has been presented [7, 54, 55], however no further ignition modelling development is reported. Good reproduction of the pressurisation rate of the Space Shuttle booster has been obtained in [54], with an inviscid flow modelling, surface heat-transfer computed via an

ad-hoc exchange law, no radiative heating, and ignition based on a simple temperature criterion. In terms of solution procedure, separate solvers are used for the internal flow field (ROCFLO and ROCFLU CFD codes), for the solid propellant deformation (ROCSOLID), for the propellant combustion (ROCBURN). These solvers may be solved with high-order in time and space separately. The complete coupled problem is solved with a number of coupling time steps, during which the solvers are executed in a staggered manner. However, the coupling variables remain constant during each coupling time step, therefore the overall coupling is only first-order accurate in time, which could impact the overall accuracy.



**Figure 1.17** Snapshot of the surface temperature distribution in the Space Shuttle SRM during ignition as simulated with ROCSTAR [154]

All the subsequent ignition modelling for 3D simulations reported in the open literature use the same simplified propellant model, with a 1D heat equation (sometimes solved more accurately using a finite-difference discretisation of the solid propellant thermal profile), and a predefined ignition temperature.

In the last decade, multiple teams of Chinese researchers have developed similar tools [8, 56–59], with a coupling of a compressible flow solver for the internal flow field and a solid mechanics solver for the propellant deformation. The 1D propellant heating model uses a finite-difference scheme with radiative contribution to the surface heating. In [56], a relatively coarse CFD mesh is used, requiring wall laws to accurately compute the heat transfer at the propellant surface. For the simulation of a dual-pulse motor [57, 58], a very fine mesh (200 million cells) is used to ensure an accurate resolution of the thermal boundary layers with a  $k-\omega$  RANS turbulence model, with cells being 1  $\mu\text{m}$  thick near the surface of the propellant. Such simulations are reported in 2D in [57] and 3D-periodic [58]. Even though the overall pressure history agrees well with experimental data, the initial pressurisation rate is overestimated. The authors attribute this to the absence of unsteady flame modelling, which could delay the pressure rise. Similarly to ROCSTAR, separate solvers are used for each problem (deformation, fluid flow, propellant combustion) and are executed sequentially in a staggered manner. Thus the overall order of accuracy in time is 1.

One of the last publications on SRM ignition is due to a collaboration between ArianeGroup and ONERA [155]. This work capitalizes on the ignition modelling developed at ONERA since the 1990s [52, 61]. The article is concerned with the reproduction of ignition in an experimental subscale motor, using the multiphysics simulation platform CEDRE from ONERA [71]. The flow field is modelled in 3D with a  $k-\omega$  RANS turbulence model, and wall laws are used to compute the convective heat flux on the propellant surface. The ignition model involves the unsteady heating of the solid propellant, with a surface pyrolysis and quasi-steady gas flame modelled via analytical relations, assuming a flame-sheet assumption. Thus the gas-phase reaction rate spatial distribution is considered as a Dirac, the position of which, relative to the propellant surface, is dependent on the pressure and surface temperature. The various parameters of this model are adjusted by a least-square optimisation method to obtain the best fit possible in terms of ignition delay, using experimental results from the laser ignition of propellant samples. Radiative heat fluxes from the

gas flame to the propellant are solved with a fully 3D approach using the CEDRE radiation solver REA. A dedicated test bench from ArianeGroup is used to validate the flame spreading, and it is shown that the numerical simulation produces good results, but that the inclusion of the radiative heat flux is of paramount importance to obtain the correct propagation velocity. Simulation of the MF test bench from ArianeGroup, a small experimental motor, yields good results overall for the pressure history during the ignition transient. The ignition simulation framework used in this work is in essence similar to ROCSTAR, however solid propellant deformation are not accounted for. In particular, the propellant combustion and fluid flow solvers are run in a staggered manner, with the coupling variables being constant across one coupling time step. Thus the overall order of accuracy in time is also 1.

### 1.5.4 Position of the present work

From the literature review we have conducted, it is clear that a 3D approach is necessary to correctly account for all geometrical peculiarities of the propellant grain and combustion chamber. In this respect, a compressible CFD code must be used to account for all phenomena, such as pressure waves and conjugate heat transfer at the propellant surface. The modelling of the propellant thermal evolution can however be conducted in one dimension, locally at each propellant-type boundary point of the CFD domain.

Most ignition models found in the literature for such an application compute the heating of the propellant as that of an inert material and, once the surface temperature reaches a predefined ignition value, the propellant boundary condition locally switches from a standard wall condition to a transpiring wall condition. However, it has been demonstrated that the ignition temperature depends on the heat flux level and the heat flux history [27, 60], with possible variations of this temperature by more than 50 K. Also, the burn rate models used after ignition usually reduce to a quasi-steady Vieille law and cannot accurately represent unsteady combustion dynamics of the solid phase.

More dynamic ignition models developed at ONERA use a simplified quasi-steady flame model, assuming an exponential shape of the gas-phase temperature profile, and a large activation energy for the single gas-phase reaction. Such a model is limited in its capability to represent various types of propellants, since some compositions are better represented with a zero activation energy [62]. Furthermore, the flame remains quasi-steady, which could impact the propellant response to pressure fluctuations.

Therefore, the present work aims at developing a new one-dimensional propellant combustion model to replace these older models. It should fit within the simulation framework used for the study of ignition at ONERA [61], acting as a dynamic boundary for the CFD toolchain CEDRE [71] that computes the chamber flow field evolution. To alleviate the above limitations, we propose to focus on a purely numerical resolution of the gas-phase flame, allowing for more complex reaction mechanisms and models to be used. As such, the combustion model will require a complete discretisation in space and time of the solid-phase and gas-phase equations, and will be comparable to detailed one-dimensional models which are specialised in the simulation of the propellant combustion. This will be discussed in detail in Chapter 2. To improve the computational efficiency, an analysis of the mathematical structure of the problem should be conducted, and then used to tailor the numerical solution strategy.

From the literature review, it also appears that all reported ignition simulation frameworks are first-order accurate in time. This can strongly impact the accuracy of the result, and may limit the

computational efficiency of the overall method. Thus, in this work, we will search for a higher-order coupling strategy, which could be introduced in an existing coupled software easily as an extension of the first-order coupled approach.

Finally, this work also aims at shedding light on some previously undiscussed issues, among which is the verification that a one-dimensional propellant flame model does not impact the solution dynamics by discarding potential interactions with the surrounding gas flow. Another issue is that of the quasi-steady flame assumption, which could have an effect on the ignition dynamics and unsteady combustion. Thus, a variant of the one-dimensional boundary model will be developed, where the gas flame is solved by the CFD tool directly, using the unsteady Navier-Stokes equations.



# Part I

## One-dimensional modelling and simulation



In the previous chapter, we have seen the role played by one-dimensional models for the simulation of solid propellant ignition. Compared to the existing large-scale ignition simulation approaches, which use simplified solid propellant models, we wish to use a model that is more generic, including the proper level of physical fidelity. Therefore in this part we focus specifically on the one-dimensional modelling of the propellant combustion and the development of the associated solver. Our aim is to produce a 1D simulation tool that is robust, efficient, accurate and that can be used either as a standalone software to simulate the ignition of various solid propellant compositions, or in a coupled manner with a CFD solver to compute the ignition transient of an SRM.

First, we study in Chapter 2 the existing models from the literature and construct a generic unsteady 1D model. We then consider in Chapter 3 a simplified model in the steady-state setting, allowing for a detailed mathematical analysis of the existence and uniqueness of the solution to be conducted. In particular, the regression speed of the surface is shown to be an eigenvalue of a nonlinear problem. A numerical method to obtain the corresponding steady-state solution profiles is constructed, which will be used in the rest of the manuscript to perform parametric studies and generate reference solutions for the verification of other simulation tools.

In Chapter 4, we go back to the generic unsteady model and present an in-depth analysis of its mathematical nature after semi-discretisation in space. It is shown that the difficulty associated with the eigenvalue in steady-state is also present in the unsteady case, in particular through the algebraic character of some of the model equations. This analysis is an important contribution as it allows for the selection of numerical methods well-suited for the simulation of the model. Historically, the mathematical nature of the system had not been clearly investigated, and this partly explains the lack of high-order solution methods, in particular in the solid propellant community.

In Chapter 5, we present the numerical strategy based on embedded stiffly accurate Runge-Kutta methods. Finally, in Chapter 6, we verify associated 1D code `VULC1D` using carefully chosen challenging test cases, clearly demonstrating its high efficiency, its versatility with respect to the physical complexity of the model, and the quality of the results.



# Chapter 2

## One-dimensional solid propellant combustion modelling

### Summary

A one-dimensional unsteady model of solid propellant combustion is presented, including a detailed gas phase representation. Particularities associated with the representation of the pyrolysis processes and of heterogeneous propellants are reviewed, and the use of the well-known ZN approach as an alternative model is discussed.

In this chapter, we present a one-dimensional modelling approach for the solid propellant combustion. This approach has been used extensively in the literature, from the first reported studies since the 1940s [2, 65] to present years [18, 33, 34]. It has enabled the study of many aspects of solid propellant combustion. Steady-state models [62, 65] have used that approach for wide variety of propellant types, matching with good accuracy the stationary regression speed. The study of the sensitivities of the latter to various physical parameters has enabled a first quantification of combustion stability [2, 30, 65] and dynamic response to pressure oscillations [94, 156, 157]. Other studies have used the one-dimensional approach for the analysis of the ignition of propellant samples [90, 153, 158, 159], yielding acceptable results in view of the available experimental data. As explained earlier (and dealt with in more details in Chapter 7), large-scale multidimensional ignition simulations of SRMs also use one-dimensional propellant models to compute the ignition of the propellant surfaces [7, 50, 56]. Indeed, that approach leads to a large reduction in the computational cost compared to the estimated cost for the numerical resolution of a hypothetical monolithic model accounting for all physical phenomena at once.

Let us recall the main physico-chemical aspects of propellant combustion, which are depicted in Figure 2.1. The solid is heated up by thermal conduction and radiation from the gas phase. In a narrow zone near its surface, the solid propellant undergoes thermal degradation (solid phase reactions yielding solid products) and decomposition (fusion and evaporation). This produces gaseous species which are injected in the gas phase. All these mechanisms will be gathered under the name “pyrolysis” for simplicity. In most cases, the solid and gas phases are separated by a superficial degradation zone, which is a transition zone where both gas and liquid species are observed, referred to as “foam” layer. This zone is usually thin, typically one micron or less [160] for ammonium perchlorate, a few dozen microns for HMX/RDX [161, 162], and its thickness decreases as pressure increases. As a consequence of this pyrolysis, the surface regresses and the injected species react

and form a flame which heats back the solid phase, allowing for a sustained surface degradation and solid phase consumption.

There are multiple reasons to choose a one-dimensional approach to represent these various phenomena. One of the main motivation is historically a practical one. It is the only approach allowing for an accessible mathematical analysis and analytical resolution in the steady regime, an advantage which was particularly attractive when computing capabilities were a strong limiting factor. Also, there exists a wide variety of so-called monopropellants, i.e. homogeneous mixtures whose burning is predominantly one-dimensional, such as AP and HMX. As research and knowledge progressed, studies started to focus on unsteady and transient phenomena such as instabilities [156, 163] and ignition [18, 158]. Detailed gas phase chemical kinetics have also emerged for steady and unsteady analysis [34, 36, 145, 164, 165]. These new research areas also favoured one-dimensional models as the computational expense was unreasonable for multidimensional simulations. That observation still holds in the current period [33, 34], the only multidimensional propellant combustion simulations being performed in 2D and steady-state with detailed kinetics [17, 166], or in unsteady 2D or 3D with simplified kinetics [19, 39, 120].

From a physical point of view, using a one-dimensional model is natural for homogeneous propellants, where the various physical fields, e.g. temperature, can clearly be considered one-dimensional due to the premixed nature of the gas flame and the homogeneity of the solid propellant. It has also been shown [33, 165] that a one-dimensional model yields good results for composite solid propellants, either in steady-state or for transient dynamics, even though the flame structure and solid phase are clearly three-dimensional as seen in Figure 1.10. Indeed, the various spatial properties and phenomena can be homogenised in the directions perpendicular to the propellant surface to form a reasonably accurate one-dimensional model, and results have been in very good agreement with experimental data [33] for a wide range of pressures. Eventually, the one-dimensional approach makes sense as the gradients (temperature, species,...) are predominantly along the perpendicular to the surface of the propellant.

Several models have been developed, with essentially two levels of description. On the one side, there exists detailed models that require advanced numerical methods, resulting in computationally expensive simulations, while on the other side analytical models may be found, which directly give simple relations to determine the steady regression speed and allow a qualitative description and global understanding of the physics at the cost of some restrictive assumptions.

Detailed one-dimensional models rely on a comprehensive set of equations to describe the various multiphysical phenomena at stake [33, 34, 94, 164], e.g. Navier-Stokes equations with species transport and reactions, radiative effects, surface kinetics and solid phase heating.

The main analytical models in steady regime assume that the pyrolysis reactions are concentrated in an infinitely narrow zone around the surface, and they mainly differ by their representation of the gaseous flame: either a flame-sheet (DBW [65] and BDP [130]) corresponding to high activation energy reaction, or a zero activation energy flame (WSB [62]). These flame assumptions allow for the steady-state solution to be derived easily, either explicitly or via simple fixed-point iterations. They mainly give the regression speed as a function of surface temperature, pressure and initial temperature of the propellant. They provide helpful qualitative insights on the relative importance of the various phenomena (pyrolysis, flame, heat diffusion...). A synthetic review of the construction of these models is given in [167].

Note that these analytical models can all be derived from the detailed ones mentioned previously, provided that the appropriate assumptions are used. This results in a great simplification of the solid and gas phase equations, enabling an analytical solution to be derived directly. However, the

ability to study unsteady dynamics and complex phenomena is lost. As we aim at developing a versatile model that is solved via numerical methods to allow more freedom in the modelling, we concentrate on the class of detailed models. Let us describe the corresponding modelling choices for each phase.

Due to the insulating nature of the solid propellant, it is observed that the thermal layer in the solid phase is very thin (typically 100  $\mu\text{m}$ ) compared to a typical propellant sample thickness (a few millimeters for a laboratory sample, up to 1 meter in an SRM), therefore the solid phase is assumed semi-infinite. In a similar fashion, the gaseous flame is typically a few hundred micrometers thick, which is much smaller than typical combustion chamber dimensions, even at laboratory scale. Therefore the gas phase is also represented as a semi-infinite domain.

The solid phase is modelled with the heat equation, and the gas phase is modelled as a reactive gas flow, using the multispecies Navier-Stokes equations. In reality, the flow velocity is very small in the gas flame, typically on the order of 1 to 10 m/s [1], such that the Mach number is negligible. Hence, a low-Mach approach can be used, where the thermodynamic pressure becomes uniform in space. Thus, pressure waves need not be resolved, leading to a simplification of the overall model. In the case of unsteady simulations with the tool presented further in Chapter 4, larger time steps will also be accessible, because the faster dynamics associated with pressure waves is discarded. We note the work of Rahman [94], where the fully compressible Navier-Stokes equation are used to study the response of the propellant to acoustic perturbations and its coupling with the mechanical deformation of the solid phase.

Both phases are separated by a superficial degradation zone near the propellant surface where heat, mass, and species fluxes are exchanged. This is a transition layer where both gas and liquid species are observed. Some detailed approaches [34, 36, 145] have a specific set of multiphase flow equations to account for its dynamics, at the cost of greatly increasing the model complexity. Additionally the experimental characterisation of the properties of this zone is out of reach of current experimental capabilities. Consequently, the vast majority of the combustion models assume that all surface phenomena are gathered within an infinitely thin layer, coincident with the propellant surface, and that these phenomena can be summarised as surface reactions that transform the solid phase into the pyrolysis products. Some studies [33, 62] consider an in-depth degradation of the solid propellant, i.e. solid phase reactions. This modelling aspect is discussed in more details in the second part of this chapter.

As mentioned in the general introduction and in Chapter 1, the one-dimensional ignition models used in large-scale simulations only consider an inert solid phase, and the gaseous flame is accounted for via empirical parameters or simple analytical models. Even if the results achieved are already of sufficient quality for engineering purposes, we believe that this approach limits the versatility of the model by imposing a restrictive and simplified modelling. On the other hand, detailed models are more accurate and can be extended easily to account for new phenomena. They are however computationally expensive, and the experimental data available is not yet able to provide sufficiently accurate measurements for the determination of their various parameters. Therefore, we choose in the present manuscript to develop an intermediate model. Based on the lack of general consensus on the modelling of the propellant degradation, and in line with previous work from ONERA [153], we do not further consider the in-depth degradation of the solid phase. Instead, the classical approach of a surface degradation pyrolysis process is considered. Note that the model can anyway easily be modified to include other degradation modelling approaches. To allow for a higher level of complexity and genericity than accessible with the analytically solved flames from the existing simplified ignition models, we choose to represent the gaseous flame as in the detailed

models with a complete set of partial differential equations. The model will require a numerical strategy to be simulated. However, we will mostly use it in association with very simplified chemical mechanisms, i.e. a global reaction mostly, therefore the associated cost will remain acceptable. The approach is fundamentally unsteady. The aim is to have a base model which will enable an accurate reproduction of ignition, while being amenable to further modelling refinements without requiring a reassessment of the solution method. The intermediate modelling level should also enable the simulation of unsteady and transient dynamics, which are of great interest in the general context of solid propulsion.

The chapter is organised as follows. First, the 1D model is derived. Then, a few specificities of the one-dimensional modelling of heterogeneous propellants are presented. The treatment of the solid phase degradation is also reviewed. Finally, a section is devoted to the well-known ZN approach, which does not require a gas phase model, and its limitations with regard to the study of ignition are discussed.

## 2.1 General modelling

In this section we present the various modelling choices classically made for the detailed modelling of the one-dimensional combustion of a solid propellant, and give the associated set of equations. The phenomenon is studied in the Galilean reference frame  $\mathcal{R}_G$ , and a schematic representation is provided in Figure 2.1. Note that the gas phase temperature profile is represented as monotonic in the figure, however there might a plateau of temperature as seen for example for HMX [147]. Also, gas phase radiation may lead to decrease of the final flame temperature.

### 2.1.1 Derivation of the model

The solid propellant combustion problem is considered in one dimension. As explained in the introduction of this chapter, it is sensible to represent both the solid and gas phases as semi-infinite mediums, which are connected via an infinitely thin interface. The interface lies at the abscissa  $\sigma(t)$ . The solid phase (propellant) is semi-infinite and is located between  $-\infty$  and  $x = \sigma(t)$ . The gas phase is also semi-infinite and is located between  $x = \sigma(t)$  and  $+\infty$ . The instantaneous signed regression speed is  $c(t) = d_t\sigma(t) < 0$ , and the absolute regression speed is  $r = -c$ .

In addition to the previous simplifications, we assume the following:

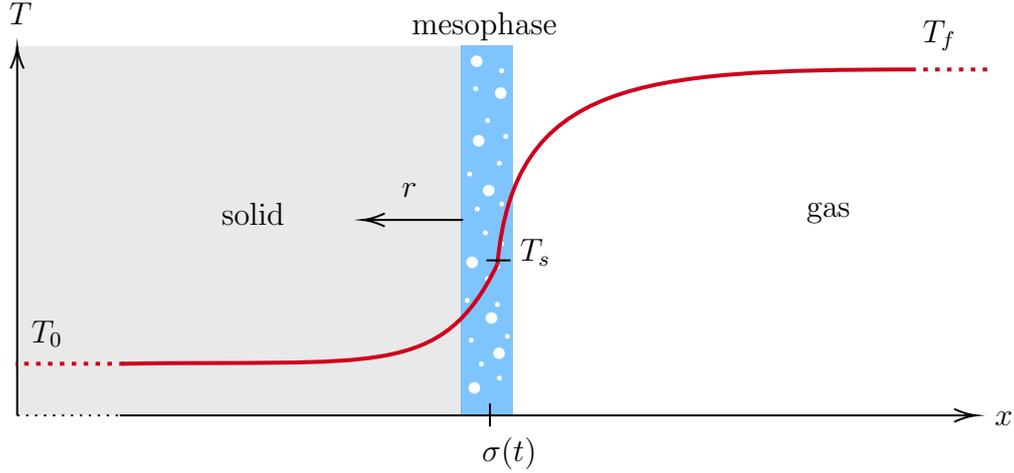
**H 1.** The solid phase is inert, incompressible and inelastic. No species diffusion takes place in the solid. Far from the burning surface, the solid phase is at its initial temperature,  $T(-\infty) = T_0$ . All gradients vanish at  $x = -\infty$ .

**H 2.** The gas phase is constituted of a mixture of reacting ideal gases in the low-Mach number limit and the thermodynamic pressure  $P$  does not vary in space.

Details on the modelling of the solid pyrolysis are discussed in Section 2.3. We introduce the following assumptions:

**H 3.** No species or heat accumulation takes place at the interface. The temperature is continuous across the interface and its value is denoted  $T_s(t)$ . The gasification process is controlled by a pyrolysis reaction concentrated at the interface. The mass flow rate of gaseous species expelled by the solid phase through the pyrolysis reaction is given by a pyrolysis law of the form:

$$m = f(T_s, P) \tag{2.1}$$



**Figure 2.1** One-dimensional model of solid propellant combustion

The pyrolysis law used in our numerical applications is  $m = A_p \exp(-T_{ap}/T_s)$ . It corresponds to a zeroth-order Arrhenius law, i.e. the reaction rate does not depend on the concentration of the reactant species. This simple law can be extended to include a dependence on pressure (typically  $P^n$ ) and an additional dependence on surface temperature ( $T_s^\beta$  with  $\beta > 0$ ) in the pre-exponential factor  $A_p$ . This law is frequently used for stationary as well as transient studies of solid propellant combustion, although it ignores some potentially important effects which only appears in more comprehensive pyrolysis relations, deduced for instance from activation energy asymptotics with zeroth-order reaction inside the solid [63]. We investigate that matter in more details in Section 2.3. All the numerical considerations made in the next chapters remain valid for any other pyrolysis law  $m$ . Note that the theoretical results from Chapter 3 are obtained with the assumption that the mass flow rate is increasing with  $T_s$ .

We also neglect radiative effects:

**H 4.** The model does not account for gas phase and surface radiation.

These simplifications greatly reduce the model complexity and are well suited for the scope of the present manuscript. Further developments are however possible by *a posteriori* including those discarded phenomena into the modelling, without requiring much change in the tools presented in this manuscript.

Finally, the diffusive processes are simplified by considering the following assumption:

**H 5.** Soret (species diffusion induced by temperature gradients) and Dufour effects (temperature diffusion induced by species concentrations gradients) are neglected in the gas phase. Species diffusive fluxes are given by a generalised Fickian law.

Note that more detailed approaches for the multicomponent diffusion are available, e.g. [168].

Using the heat equation to model the evolution of the temperature inside the solid, and the low-Mach limit of the Navier-Stokes equations [169] with reactions and species transport for the gas phase allows us to describe the evolution of our system.

The solid phase is represented by its temperature  $T(x, t)$  and its constant density  $\rho_c$  for  $x < \sigma(t)$ . The gas phase is described by the density  $\rho(x, t)$ , the uniform pressure  $P(t)$ , the flow speed  $u(x, t)$ , and the temperature  $T(x, t)$  for  $x > \sigma(t)$ .

## Reactive aspects

The reactive aspect of the flow with  $n_e$  species (symbol  $\mathcal{E}_\mathfrak{k}$ ,  $\mathfrak{k} \in \llbracket 1, n_e \rrbracket$ ) is taken into account with the addition of the transport equations for the species mass fractions  $Y_\mathfrak{k}(x, t)$  and the addition of the volumetric heat release as a source term in the energy equation. The mass production rate of the  $\mathfrak{k}$ -th species per unit volume is  $\omega_\mathfrak{k}$ , in  $\text{kg} \cdot \text{m}^{-3} \cdot \text{s}^{-1}$ . We consider  $n_r$  chemical reactions of the form:



We introduce  $\nu_{\mathfrak{k},r} = \nu''_{\mathfrak{k},r} - \nu'_{\mathfrak{k},r}$ , the global stoichiometric coefficient. The reaction rate of the  $r$ -th reaction is  $\tau_r$ , in  $\text{mol} \cdot \text{m}^{-3} \cdot \text{s}^{-1}$ . It is typically a generalized Arrhenius law dependent on species concentrations, temperature and pressure. We have the relation  $\omega_\mathfrak{k} = \mathcal{M}_\mathfrak{k} \sum_{r=1}^{n_r} \nu_{\mathfrak{k},r} \tau_r$ , with  $\mathcal{M}_\mathfrak{k}$  the molar mass of this species.

The heat release generated by the chemical reactions are directly obtained via conservation of the mixture enthalpy  $h$ , which is the sum of the chemical and sensible enthalpies and is expressed in  $\text{J} \cdot \text{kg}^{-1}$ :

$$h = \sum_{\mathfrak{k}=1}^{n_e} Y_\mathfrak{k} h_\mathfrak{k} \quad (2.3)$$

where the enthalpy of the  $\mathfrak{k}$ -th species is:

$$h_\mathfrak{k} = h_\mathfrak{k}^0 + \int_{T_{std}}^T c_{p,\mathfrak{k}}(a) da \quad (2.4)$$

with  $c_{p,\mathfrak{k}}$  the heat capacity of the  $\mathfrak{k}$ -th species, and  $h_\mathfrak{k}^0$  its formation enthalpy at the standard temperature  $T_{std}$ . The molar enthalpy of the  $\mathfrak{k}$ -th species in  $\text{J} \cdot \text{mol}^{-1}$  is:

$$h_{i,mol}(T) = \mathcal{M}_\mathfrak{k} h_\mathfrak{k} \quad (2.5)$$

## Surface species

The gas and solid phases are coupled at the interface through boundary conditions obtained by integration of the energy and transport equations around the interface. Assuming the pyrolysis process is concentrated at the interface, we introduce the “injection” mass fractions  $Y_{\mathfrak{k},inj} = Y_\mathfrak{k}(\sigma^-)$  for the different gaseous species, which indicates the mass fractions obtained after pyrolysis directly at the interface, before entering the gas phase.

## Surface tracking

Finally, we perform a variable change, such that the interface remains at a constant abscissa, as described in [94]. We introduce the new space variable  $\hat{x} = \hat{x}(x, t) = x - \int_0^t c(\eta) d\eta$ , where  $c(t) = -r(t)$  is the instantaneous regression velocity of the interface. In the Galilean reference frame  $\mathcal{R}_G$ , the interface lies at the abscissa  $\sigma(t) = \sigma(0) + \int_0^t c(\eta) d\eta$ , therefore following our variable change, the new interface abscissa is  $\hat{\sigma}(t) = \sigma(0)$ , which is constant in time. We assume, without any loss of generality, that this position is 0.

If, for a function  $f(x, t)$ , we introduce  $\hat{f}(\hat{x}, t)$  such that  $\hat{f}(\hat{x}(x, t), t) = f(x, t)$ , we can derive the following relations:

$$(\partial_x f)_t = (\partial_{\hat{x}} \hat{f})_t \quad (\partial_t f)_x = -c(t) (\partial_{\hat{x}} \hat{f})_t + (\partial_t \hat{f})_{\hat{x}} \quad (2.6)$$

Therefore, the change of variable introduces additional convective terms in the equations. The regression velocity of the propellant surface is  $-r < 0$ . For the sake of simplicity, we drop the “ $\hat{\cdot}$ ” notation in the rest of the chapter.

## 2.1.2 The complete model

Following the previous modelling choices, we can now write down the constitutive equations of our model.

### Solid phase

The solid phase temperature field  $T$  at  $x < 0$  is subject to:

$$\rho_c c_c \partial_t T + \rho_c c_c r \partial_x T - \partial_x (\lambda_c \partial_x T) = 0 \quad (2.7)$$

with  $\rho_c$  the propellant density,  $c_c$  its heat capacity,  $\lambda_c$  the thermal conductivity, and  $r$  the absolute surface regression speed, deduced from Equation (2.25) in the next section. Far below the surface, the solid is at its resting temperature:

$$T(-\infty) = T_0 \quad (2.8)$$

### Gas phase

The flame at  $x > 0$  is modelled as a low-Mach reactive flow, subject to the following partial differential equations:

$$\begin{cases} \partial_t \rho + \partial_x (\rho(u + r)) = 0 & (2.9) \\ \partial_t \rho Y_{\mathfrak{k}} + \partial_x (\rho(u + r) Y_{\mathfrak{k}}) = -\partial_x J_{\mathfrak{k}} + \omega_{\mathfrak{k}} & \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket & (2.10) \\ \partial_t \rho h + \partial_x (\rho(u + r) h) = \partial_t P - \partial_x (-\lambda \partial_x T + \sum_1^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k}}) & (2.11) \end{cases}$$

where  $\rho$  denotes the density,  $Y_{\mathfrak{k}}$  the mass fraction of the  $\mathfrak{k}$ -th species,  $u$  the flow velocity. The thermal conductivity is  $\lambda$ , and the thermodynamic pressure is  $P$ . The mixture is composed of  $n_e$  species. The volumetric production rate of the  $\mathfrak{k}$ -th species is  $\omega_{\mathfrak{k}}$ . The diffusion flux  $J_{\mathfrak{k}}$  of the  $\mathfrak{k}$ -th species is evaluated with a variant of the Hirschfelder-Curtiss approximation [170, 171]. It relates the diffusive flux of one species to the gradient of its mass fraction:

$$J_{\mathfrak{k}} = \rho Y_{\mathfrak{k}} V_{\mathfrak{k}} = -\rho D_{\mathfrak{k}}^{eq} \partial_x Y_{\mathfrak{k}} \quad (2.12)$$

where  $V_{\mathfrak{k}}$  is the  $\mathfrak{k}$ -th species diffusion velocity, and the equivalent diffusion coefficient  $D_{\mathfrak{k}}^{eq}$  is defined as:

$$D_{\mathfrak{k}}^{eq} = \frac{\sum_{j \neq \mathfrak{k}} X_j}{\sum_{j \neq \mathfrak{k}} X_j / D_{j\mathfrak{k}}} \quad (2.13)$$

where the  $D_{\mathfrak{k}j}$ 's are the components of the species diffusion matrix, which is symmetric. The molar fraction of the  $\mathfrak{k}$ -th species is defined as:

$$X_{\mathfrak{k}} = \frac{\mathcal{M}}{\mathcal{M}_{\mathfrak{k}}} Y_{\mathfrak{k}} \quad (2.14)$$

with the mean molecular weight  $\mathcal{M}$  defined as:

$$\frac{1}{\mathcal{M}} = \sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}} \quad (2.15)$$

To ensure mass conservation, it is required that  $\sum_{\mathfrak{k}=1}^{n_e} J_{\mathfrak{k}} = 0$ , otherwise summing up Equation (2.10) for all species does not exactly yield the continuity equation (2.9). The previous formulation usually does not verify this property. Therefore, it is common practice to correct the diffusive fluxes by introducing a corrective diffusion velocity  $V_c$  in the flux formulation:

$$J_{\mathfrak{k}} = \rho Y_{\mathfrak{k}}(V_{\mathfrak{k}} + V_c) \quad (2.16)$$

This velocity is computed as:

$$V_c = \sum_{\mathfrak{k}=1}^{n_e} -Y_{\mathfrak{k}} V_{\mathfrak{k}} \quad (2.17)$$

With this correction, it is straightforward to find that the species diffusive fluxes now sum up to zero.

The mixture-averaged thermal conductivity  $\lambda$  is obtained as the average of the arithmetical and harmonic means of the species conductivities weighted by their mole fractions:

$$\lambda = \frac{1}{2} \left( \sum_{\mathfrak{k}=1}^{n_e} X_{\mathfrak{k}} \lambda_{\mathfrak{k}} + \frac{1}{\sum_{\mathfrak{k}=1}^{n_e} X_{\mathfrak{k}} / \lambda_{\mathfrak{k}}} \right) \quad (2.18)$$

The ideal gas law relates the various state variables in the gas phase:

$$\rho = P / \left( RT \sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}} \right) \quad (2.19)$$

with  $R$  the universal gas constant.

The gas phase being semi-infinite, equilibrium is ensured to be reached after a certain abscissa, therefore the exit boundary conditions for the 1D flame are simple Neumann conditions:

$$\partial_x T(+\infty) = 0, \quad \partial_x Y_{\mathfrak{k}}(+\infty) = 0 \quad \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket \quad (2.20)$$

## Surface connection conditions

The fluid and solid models are connected through an infinitely thin interface which is coincident with the surface of the solid propellant. We denote the surface variables by the subscript  $s$ , i.e. the values of the various fields at  $x = 0$ . We include the contribution of an external stimulus, e.g laser heat flux, through the term  $q_{ext}$ .

The connection of the gas and solid phases at the interface is given by the following conditions, expressing the continuity of the mass flow rate and temperature, as well as the enthalpy and species fluxes balance around the interface:

$$\left\{ \begin{array}{l} \rho_c r = m(0^+) \end{array} \right. \quad (2.21)$$

$$\left\{ \begin{array}{l} T(0^-) = T(0^+) = T_s \end{array} \right. \quad (2.22)$$

$$\left\{ \begin{array}{l} (mh - \lambda_c \partial_x T)_{0^-} = (mh - \lambda \partial_x T + \sum_1^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k}})_{0^+} - q_{ext} \end{array} \right. \quad (2.23)$$

$$\left\{ \begin{array}{l} (mY_{\mathfrak{k},inj})_{0^-} = (mY_{\mathfrak{k}} + J_{\mathfrak{k}})_{0^+} \quad \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket \end{array} \right. \quad (2.24)$$

$m$  the mass flow rate ( $\rho_c r$  in the solid,  $\rho u$  in the gas) and  $Y_{inj}$  the product mass fractions generated by the decomposition and gasification processes, which in our applications will be user-specified constants. Note that they could also be functions of the surface temperature as in [119]. Equation (2.24) is the species balance, i.e. the flow rate of the  $\mathfrak{k}$ -th species generated by the pyrolysis is equal to the flow rate of this species leaving the surface in the gas phase, minus the species diffusion flow rate.

Finally, all the complex surface phenomena responsible for the depletion rate of the solid propellant are described by a pyrolysis law which dictates the surface mass flow rate:

$$m(0) = m_s = \rho_c r = f(T_s, P) \quad (2.25)$$

## Remarks

A few remarks may be useful. First, in this one-dimensional framework, the momentum equation is redundant, as the mass flow rate spatial variation is already determined by the continuity equation from the  $\rho$  temporal evolution, which itself is known from the temporal variations of  $T$  and  $Y_{\mathfrak{k}}$  and the thermodynamic pressure  $P$  through the equation of state (2.19). Hence  $\rho$ , although its time derivative is specified by the continuity equation, is not a true variable. As we will see in Section 4.3, the continuity equation only acts as a constraint that determines the velocity field. The momentum equation could be used to determine the hydrodynamic pressure field, however it is known that this pressure perturbation is of the order of  $\text{Ma}^2$ , with  $\text{Ma}$  the Mach number. In solid propellant flames, the Mach number is typically of the order of  $10^{-3}$ , hence the pressure perturbation can be completely neglected. For two- or three-dimensional low-Mach flows, the continuity equation does not allow for the determination of the multiple components of  $u$ , therefore the momentum equation is required.

Second, the variable change we have performed is not a change of reference frame, therefore no inertial body forces appear. Indeed the gas-phase velocity  $u$  is still the one observed in the original Galilean reference frame. We see that all the convective terms involve  $(u + r)$  and  $u$  never appears on its own. We simplify the notations by introducing the gas mass flow rate  $m = \rho(u + r)$ . Note that in all practical cases the regression speed is lower 1% of the gas phase velocity, therefore  $m$  is very close to  $\rho u$ .

Third, all the partial differential equations presented above are in conservative form, which we will rely on for a finite volume discretisation later on in Chapter 4. However, an alternative formulation may be useful for theoretical analysis, in particular the conservation of enthalpy can be replaced by the equivalent equation on the temperature:

$$\rho c_p \partial_t T + m c_p \partial_x T - \partial_x (\lambda \partial_x T) + \sum_{\mathfrak{k}=1}^{n_e} J_{\mathfrak{k}} \partial_x h_{\mathfrak{k}} = d_t P - \sum_{\mathfrak{k}=1}^{n_e} h_{\mathfrak{k}} \omega_{\mathfrak{k}} \quad (2.26)$$

The interface thermal balance equation (2.23) can also be replaced by an equivalent equation for the temperature  $T$ :

$$(\lambda_c \partial_x T)_{0-} = m Q_p + (\lambda \partial_x T)_{0+} + q_{ext} \quad (2.27)$$

with  $Q_p$  the heat of reaction associated with the pyrolysis and gasification processes (surface reactions), which depends on the temperature  $T_s$  and surface mass fractions  $Y_{s,\mathfrak{k}}$ . From a physical point of view, Equation (2.27) means that the heat conducted from the gas phase into the solid and the heat generated by the pyrolysis process (when the pyrolysis is exothermic) are used to heat up the solid propellant and sustain the combustion.

## 2.2 1D modelling of heterogeneous propellants

The previous general model is especially suited to the representation of homogeneous propellants, such as AP or HMX. However, it has been shown that a one-dimensional model can be very accurate for the burning of composite propellants such as AP-HTPB for pressures up to 5 MPa [33], i.e. in the range of pressures typically encountered in a solid rocket motor. In that case however, the thermal properties of the composite propellants have to be homogenised to obtain a meaningful one-dimensional equivalent. Let us study that point in more details, first for the specific heat and the density and, in a second section, for the thermal conductivity, with a specific focus on AP-HTPB formulations.

### 2.2.1 Specific heat and density

Let us consider a mixture of AP-HTPB of mass  $m_{tot}$  and volume  $V_{tot}$ , containing a mass  $m_{AP}$  of AP. The mass fraction of AP is thus  $\alpha = m_{AP}/m_{tot}$  and typical solid propellants have on  $\alpha \approx 0.8$ . Let  $\rho_k, f_k, c_{p,k}, h_k^0$  the density, volume fraction, heat capacity, formation enthalpy, of the components and of the mixture of the component  $k$  which is either  $AP, H$  (for HTPB) or  $tot$  for the mixture.

Let us determine the density of the mixture. The volume of the mixture is the sum of the volumes of AP and HTPB:

$$V_{tot} = V_{AP} + V_H \quad (2.28)$$

$$\Rightarrow V_{tot} = \frac{m_{AP}}{\rho_{AP}} + \frac{m_H}{\rho_H} = \frac{\alpha m_{tot}}{\rho_{AP}} + \frac{(1-\alpha)m_{tot}}{\rho_H} \quad (2.29)$$

Therefore the mixture density is:

$$\rho_{tot} = \frac{m_{tot}}{V_{tot}} = \frac{1}{\frac{\alpha}{\rho_{AP}} + \frac{1-\alpha}{\rho_H}} \quad (2.30)$$

i.e. it is the harmonic mean of the components densities, with weights equal to their respective mass fractions.

The volume fraction of AP can then be computed as:

$$f_{AP} = \frac{V_{AP}}{V_{tot}} = \frac{m_{AP}}{\rho_{AP}} \frac{1}{V_{tot}} = \frac{m_{AP}}{\rho_{AP}} \frac{\rho_{tot}}{m_{tot}} = \alpha \frac{\rho_{tot}}{\rho_{AP}} \quad (2.31)$$

$$\Rightarrow f_{AP} = \frac{1}{1 + \frac{\rho_{AP}}{\rho_H} \frac{1-\alpha}{\alpha}} \quad (2.32)$$

Let us now focus on the derivation of the mixture specific heat  $c_{p,tot}$ . This can be simply determined by considering the sensible enthalpy of the mixture, and using the property of extensivity of the enthalpy. We can write:

$$m_{tot} \left( h_{tot}^0 + \int c_{p,tot}(T) dT \right) = m_{AP} \left( h_{AP}^0 + \int c_{p,AP}(T) dT \right) + m_H \left( h_H^0 + \int c_{p,H}(T) dT \right) \quad (2.33)$$

By deriving the previous enthalpy equation with respect to temperature, we get:

$$\Rightarrow m_{tot} c_{p,tot}(T) = m_{AP} c_{p,AP}(T) + m_H c_{p,H}(T) \quad (2.34)$$

Therefore, dividing by  $m_{tot}$ , we find:

$$c_{p,tot}(T) = \alpha c_{p,AP}(T) + (1 - \alpha) c_{p,H}(T) \quad (2.35)$$

And the standard mixture enthalpy can be simply obtained:

$$h_{tot}^0 = \alpha h_{AP}^0 + (1 - \alpha) h_H^0 \quad (2.36)$$

## 2.2.2 Effective thermal conductivity

The determination of the effective thermal conductivity of the heterogeneous mixture is a more complicated matter. There have been multiple approaches to determine an effective thermal conductivity, based on the volume fractions and thermal conductivities of each component [128, 172]. Here, we briefly detail the computation of the two simplest approaches, and then compare the results obtained for a conventional AP-HTPB propellant with other homogenisation strategies.

### Parallel mode

In this mode, we consider that the propellant forms a sandwich and that heating occurs predominantly parallel to the layers, with the layers sides exposed to the same temperature. The case is sketched in Figure 2.2a. We suppose the sandwich is made of a superposition of  $2N$  layers, half of which are AP and the other half is HTPB. The overall thickness is  $H$ .

So as to maintain the desired volume fraction, the  $k$ -th layer has a thickness  $h_k = H \frac{f_k}{N(f_{AP} + f_H)} = hf_k/N$ ,  $f_k$  being chosen as  $f_{AP}$  or  $f_H$  depending on the nature of the layer. Thus, the ratio of the surface of AP exposed to the side divided by the overall side surface is equal to the volume fraction  $f_{AP}$ . We impose the temperature at the top and bottom of the sandwich to  $T_u$  and  $T_d$  respectively. We neglect lateral heat transfer. In steady-state, the overall heat flux that goes through the sandwich is the sum of the fluxes through each layer:

$$\Phi = f_{AP} \lambda_{AP} \frac{T_u - T_d}{H} + (1 - f_{AP}) \lambda_H \frac{T_u - T_d}{H} \quad (2.37)$$

With an effective thermal conductivity  $\lambda_{tot}$ , we can write:

$$\Phi = \lambda_{tot} \frac{T_u - T_d}{H} \quad (2.38)$$

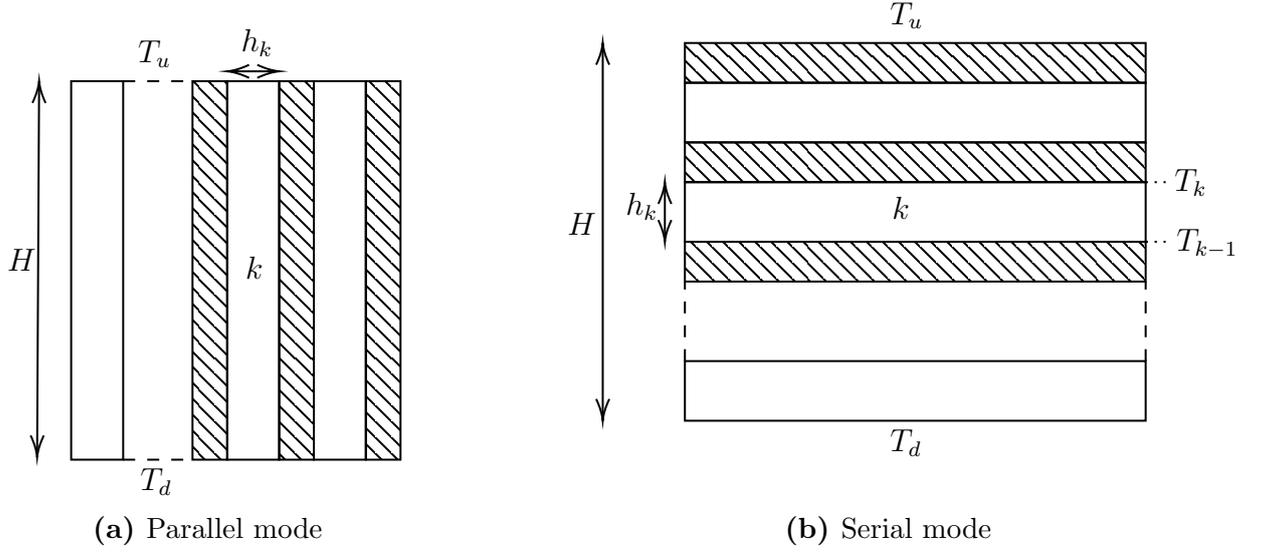
Therefore, equating the two previous expressions leads to:

$$\lambda_{tot} = f_{AP} \lambda_{AP} + (1 - f_{AP}) \lambda_H \quad (2.39)$$

i.e. the effective thermal conductivity is the arithmetical mean of those of each component.

### Serial mode

In that mode, the sandwich is rotated by 90 degrees, such that the temperature gradient goes through each layer successively. The case is sketched in Figure 2.2b. We neglect lateral heat fluxes which could cause heat loss to the sides. Let  $T_k$  the temperature at the boundary between the  $k$ -th and  $(k + 1)$ -th layer. We impose  $T_N = T_u$  and  $T_0 = T_d$ . In steady-state, the heat equation yields  $d_{xx}T = 0$  in each layer,  $x$  being the direction perpendicular to the layers.



**Figure 2.2** Two simple homogenisation approaches for the determination of the effective thermal conductivity

The temperature gradient in the  $k$ -th layer is  $\frac{T_k - T_{k-1}}{h_k}$ . We can write the following equation:

$$T_u = T_d + \sum_{k=1}^{2N} (T_k - T_{k-1}) \quad (2.40)$$

At steady-state, the heat flux must be uniform in space, thus the heat flux in each layer is equal to the overall heat flux expressed with the effective thermal diffusivity of the sandwich:

$$\lambda_k \frac{T_k - T_{k-1}}{h_k} = \lambda_{tot} \frac{T_u - T_d}{H} \quad (2.41)$$

$$\Rightarrow T_k - T_{k-1} = \frac{h_k \lambda_{tot}}{\lambda_k H} (T_u - T_d) \quad (2.42)$$

Thus, Equation (2.40) can be rewritten as:

$$T_u - T_d = \sum_{k=1}^{2N} \frac{h_k \lambda_{tot}}{\lambda_k H} (T_u - T_d) \quad (2.43)$$

Therefore:

$$\lambda_{tot} = \frac{H}{\sum_{k=1}^{2N} \frac{h_k}{\lambda_k}} = \frac{1}{N \frac{f_{AP}}{\lambda_{AP}} + N \frac{f_H}{\lambda_H}} = \frac{1}{\frac{f_{AP}}{\lambda_{AP}} + \frac{f_H}{\lambda_H}} \quad (2.44)$$

i.e. the effective thermal conductivity is the harmonic mean of the components conductivities.

### 2.2.3 Other approaches

A wide variety of homogenisation methods have been developed for porous or heterogeneous materials and may be applied to the solid propellant case. Of particular interest are the methods that

$\rho_{AP}$	$\rho_H$	$\lambda_{AP}$	$\lambda_H$	$\alpha$
1950 kg.m <sup>-3</sup>	920 kg.m <sup>-3</sup>	0.405 W.m <sup>-1</sup> .K <sup>-1</sup>	0.276 W.m <sup>-1</sup> .K <sup>-1</sup>	0.86

**Table 2.1** Input data

$f_{AP}$	0.79	$\lambda_{tot}$ (parallel)	0.378 W.m <sup>-1</sup> .K <sup>-1</sup>
$\rho_{tot}$	1734 kg.m <sup>-3</sup>	$\lambda_{tot}$ (Eucken)	0.375 W.m <sup>-1</sup> .K <sup>-1</sup>
$\lambda_{tot}$ (serial)	0.369 W.m <sup>-1</sup> .K <sup>-1</sup>	$\lambda_{tot}$ (EMT)	0.375 W.m <sup>-1</sup> .K <sup>-1</sup>

**Table 2.2** Homogenised for properties for an 86 % AP - 14 % HTPB mixture

consider the case of particles of one material held within a matrix of different nature, for example the Maxwell-Eucken formula [173], or the effective medium theory (EMT) that considers spherical inclusions of a component inside a matrix of a second component [174], and also the work specifically adapted to composite solid propellants by Chen et al. [128].

Also, the homogenised properties can be determined by direct numerical simulations (DNS), as used in [128] for validation of their analytical formula. The DNS simulates the heating of a pack of AP particles held within a matrix of HTPB. Steady-state considerations of a periodic domain with known boundary heat fluxes can be used to compute a numerical effective thermal conductivity.

Last, experimental apparatus can be useful to determine the mixture-averaged properties. For example, a diffusimeter can be used to measure the effective thermal conductivity, as later detailed in Chapter 9. This will be the chosen approach in this work, as laboratory measurements are directly available for the propellants studied. This method is however not applicable to the study of theoretical formulations, which are of particular interest for simulation frameworks which aim at providing a means of numerically optimising a propellant composition, e.g. [120].

### 2.2.4 Example result for a typical AP-HTPB blend

To demonstrate the previous formulas, we briefly give some results of the homogenisation of a mixture composed at 86 % of PA and 14 % of HTPB. The thermal properties of AP and HTPB are taken from [120] and are listed in Table 2.1. Table 2.2 gives the main results.

The parallel and serial modes usually form the upper and lower bounds for the effective conductivities computed with other formulae. In this case, these bounds are 2 % apart, indicating the very good agreement of all the homogenisation formulae for that particular composition. The choice of the homogenisation procedure will actually be more decisive if components with vastly different properties (for instance aluminium) are added [148].

## 2.3 Pyrolysis laws

In this section, we briefly discuss the representation of the pyrolysis and gasification processes. In the one-dimensional model exposed in this chapter, we have discarded in-depth degradation of the solid phase.

Experimentally, the steady-state burning rate  $\bar{r}$  can be measured via closed bomb tests, yielding a law of the form  $\bar{r} = aP^n$ ,  $a$  and  $n$  being constants, or potentially piecewise-constant functions of the pressure  $P$ . Note that the coefficient  $a$  typically depends on the initial temperature of the solid propellant.

For transient simulations of the burning of a solid propellant, the unsteady burning rate must be known. For many such applications, defining the burn rate as a function of the pressure such as the previous Vieille law is not acceptable. In the case of established burning, solid phase inertia may have a strong effect if the characteristic time of the physical fluctuations (chamber pressure or others) is on the order or lower than the solid phase characteristic time  $\tau_c = D_{th,c}/\bar{r}^2$ , with  $D_{th,c} = \lambda_c/(\rho_c c_c)$  the thermal diffusivity of the solid phase. This characteristic time is typically on the order of 3 ms, such that unsteady fluctuations (pressure or others) at frequencies above 100 Hz may already lead the solid propellant combustion into a non-quasi-steady behaviour. Finally, a steady-state law is only acceptable when established burning has been reached. In particular, it is irrelevant for the simulation of highly nonlinear transients such as ignition, where strong regression speed variations occur at near constant pressure.

Therefore, the approach universally taken is to assume that an additional relation exists, which relates the unsteady surface temperature  $T_s$  to the unsteady burning rate  $r$ . This law is referred to as the pyrolysis law and corresponds to Equation (2.25). Steady-state surface temperature measurements are difficult and suffer from some imprecision, hence they cannot be used to generate such a law. The approach universally taken is therefore to choose a theoretical formula for the pyrolysis, typically one of those presented further in this Section, and to fit its parameters to the measurements in a least-square manner.

### 2.3.1 In-depth degradation kinetics

To better investigate that aspect, let us consider the modelling of the solid propellant degradation. The solid phase may decompose beneath its surface due to the high temperature, a phenomenon which is referred to as distributed pyrolysis, and has often been suggested as an important mechanism for the combustion of solid substances [175–179]. In that case the solid phase can be modelled as a system where heat diffusion and chemical reactions occur. As a basic model, we can introduce the distributed pyrolysis reaction:



i.e. it transforms the solid propellant component  $P$  at solid state into the pyrolysis products at solid state. The coefficient  $\gamma$  is used to ensure conservation of the overall mass. As the products pass through the surface, they are instantaneously gasified.

Accounting for this reaction requires the tracking of the profile of  $Y_P$  inside the solid phase, hence a species transport equation is required, as in the gas phase. However, this equation is simplified by assuming that no species diffusion takes place inside the solid phase, as in most such models [18]. Indeed, molecular diffusion coefficients in solids are several orders of magnitude smaller than the thermal diffusivity [180, 181].

Thus the solid phase system is transformed to the following [163, 182]:

$$\begin{cases} \partial_t Y + r \partial_x \rho_c Y_p = -\Omega & (2.46) \\ \partial_t \rho h + \partial_x (\rho(u+r)h) = \partial_x (\lambda \partial_x T) & (2.47) \end{cases}$$

the reaction rate  $\Omega$  is usually taken as a zeroth-order reaction:

$$\Omega = \rho_c A_c \exp\left(-\frac{E_c}{RT}\right) \quad (2.48)$$

The species equation in the solid is associated with the boundary condition:

$$Y(-\infty, t) = 1 \quad (2.49)$$

The surface of the propellant is then defined as the abscissa  $\sigma$  such that  $Y(\sigma, t) = 0$ , i.e. the degradation is complete. Assuming without any loss of generality that  $\sigma = 0$  following our variable change, integrating Equation (2.46) from  $x = -\infty$  to  $x = 0$  leads to the following formula for the pyrolysis mass flow rate  $m$ :

$$m(t) = \int_{-\infty}^0 \Omega(x, t) dx + \rho c \int_{-\infty}^0 \partial_t Y_P(x, t) dx \quad (2.50)$$

### 2.3.2 Asymptotic expansion for the distributed reaction

In the case where the activation energy  $E_c$  is large, the reactive layer is very thin and an asymptotic expansion of the solution can be performed [163, 175, 178, 183], also known as activation energy asymptotics (AEA), assuming the degradation layer to be quasi-steady. The solid phase is split into two phases, one where reactive effects dominate, and one where only convection and diffusion can be accounted for. Solving each part separately and connecting them leads to an implicit formula for the pyrolysis mass flow rate [163]:

$$m^2 = \frac{A_c \rho_c \lambda_c T_s^2 R \exp(-E_c/RT_s)}{E_c \left( \frac{\lambda_c}{m} \partial_x T(0^-) - \frac{Q_c}{2} \right)} \quad (2.51)$$

with  $Q_c$  the overall heat-release due to the in-depth degradation. A slightly reformulated alternative is:

$$m = T_s \exp\left(\frac{-E_c}{2RT_s}\right) \sqrt{\frac{A_c \rho_c \lambda_c R}{E_c \left( \frac{\lambda_c}{m} \partial_x T(0^-) - \frac{Q_c}{2} \right)}} \quad (2.52)$$

First-order reaction rates may also be used in the condensed phase, however results have not shown any benefit from this added dependency [178].

### 2.3.3 Surface reaction

The consideration that the pyrolysis processes are concentrated at the surface of the propellant has also led to a simpler quasi-steady surface reaction model that considers a global surface reaction, with a zeroth-order rate, resulting in the following pyrolysis law:

$$m = A_s P^{n_s} T_s^{\beta_s} \exp\left(\frac{-E_s}{RT_s}\right) \quad (2.53)$$

The quasi-steadiness is related to the fact that no accumulation of reactant occurs, and that the surface processes are tacitly assumed inertialess. This surface reaction formalism has been extended by Rahman [94] to the case of complex surface kinetics. Often, the parameters  $\beta_s$  and  $n_s$  are simply set to 0.

With this classical pyrolysis law, the heat release due to the solid phase degradation is concentrated at the surface, whereas the distributed reaction model spreads it within the reaction layer.

This concentration has the effect of increasing the temperature gradient below the surface, which may have a noticeable impact on the unsteady dynamics [182].

Compared to the AEA solution (2.51), this equation includes, even if only in an ad-hoc manner, a direct effect of the gas pressure  $P$  on the surface kinetics, whereas the zeroth-order in-depth degradation (pressure-independent reaction rate) does not. With the distributed pyrolysis, pressure effects are only conveyed via the gas flame heat feedback, which has an impact on  $\partial_x T(0^-)$ .

### 2.3.4 Shortcomings

The previous models, though they have merits in their relative fidelity and ease of use, also have strong limitations. While some experiments have shown that certain compositions can indeed be considered to have a bijection between their surface temperature and the pyrolysis mass flow rate [184], thermocouple measurements for pure AP have hinted that this could not be the case for pure AP [185]. The AEA solution (2.52) has the merit that, via the term  $\partial_x T(0^-)$ , this behaviour can be at least qualitatively reproduced.

Even though this solution has been praised for its adequacy in steady-state, investigations of its unsteady behaviour has shown that it produces high-frequency artefacts in nonlinear transients, e.g. pressurisation phase of a combustion chamber [163]. Furthermore, the AEA model (2.52) was initially praised in [186] for its better realism in terms of sensitivity of the mass flow rate to the various inputs (initial temperature, pressure...) over the surface model (2.53). However it was later shown by the same authors [163] that both models can be made equivalent by carefully setting their parameters. Note that in that case, the surface model often requires parameter values that may seem partially out of physically expected bounds. Much work has also been performed by De Luca and co-workers to study the sensitivities of the classical pyrolysis law and of the AEA law and their relation to experimental data [187], showing that the various pyrolysis models all have their own limitations and affect the stability of the steady-state combustion, therefore also the unsteady behaviour.

Another issue is that of the assumption of quasisteadiness for the solid phase degradation process. This has been shown to affect the unsteady response of the surface mass flow rate to pressure fluctuations significantly, even at low frequencies and in the linear domain, compared to the unsteady decomposition process considered in Section 2.3.1 [182, 188]. In steady-state, the AEA can be in very good agreement with the complete in-depth degradation model, however the agreement may become much worse under unsteady conditions.

The determination of the true chemical pathways for actual propellants is also complex and the mechanisms are difficult to assess [160, 161]. Experimental observations are complex and do not yield sufficient accuracy so far. Even more complex is the modelling of the multiphase foam layer above the surface [33]. That is one of the reasons why many models, including multidimensional ones [19, 148] do not account for in-depth degradation effects but instead rely on the classical pyrolysis law approach.

For composite propellants, matters are even more complicated, as multiple materials undergo pyrolysis. Hence, in a one-dimensional model, there might be a need to homogenise the pyrolysis laws of the various ingredients. Chen et al. [128] have shown that a simple harmonic mean of each component's pyrolysis law yielded a good accuracy compared to results obtained with a precise simulation of a heterogeneous propellant combustion. Alternatively, they also empirically derived a formulation to generate an equivalent mixture pyrolysis law based on the laws of each component.

Following the experience from ONERA, we have chosen to employ the classical surface pyrolysis

formula. Its parameters will be numerically fitted to experimental results, see Chapter 9. Once again, other approaches are easily incorporated in the simulation tools presented further in this manuscript.

## 2.4 The ZN approach

The previous detailed model, as well as all the other models mentioned so far, rely on the use of a flame model. This approach is commonly referred to as *flame modelling* approach (FM) and has been widely used in the American and European communities since the 1960s [65, 156]. The Russian community had initially taken a vastly different approach, now referred to as the Zeldovich-Novozhilov approach (ZN), great reviews of which are available [30, 189]. This theory was originally developed to avoid relying on a flame model for the gas-phase phenomena, only modelling the solid phase in an identical manner as presented in Section 2.1.

In the FM approach, the gas flame model is only required to provide the gas flame heat feedback  $(\lambda\partial_x T)(0^+)$ , i.e. the heat that is conducted into the surface of the propellant by the flame. In the ZN approach, the approach is different. It relies on the fact that steady-state laws are available for the pyrolysis mass flow rate  $m$  and the surface temperature  $T_s$ , with the following form:

$$m = \bar{m}(T_0, P), \quad T_s = \bar{T}_s(T_0, P) \quad (2.54)$$

where  $T_0$  is the temperature deep in the solid phase, which we also refer to as initial temperature. Such relations can be obtained via theoretical modelling (FM approach) or numerical investigations for solid propellant models, or via experiments (original ZN approach) for real propellants.

The ZN approach consists in assuming that the steady-state relations (2.54) remain valid under unsteady conditions. However,  $T_0$  is generally a constant and, if  $P$  is also constant, these relations would result in a constant surface temperature and regression speed. To circumvent this issue, let us consider the steady-state profile temperature profile in the solid phase:

$$T(x) = T_0 + (T_s - T_0) \exp(xm c_c / \lambda_c) \quad (2.55)$$

The temperature gradient just below the surface is:

$$\phi = \partial_x T(0^-) = m c_c (T_s - T_0) / \lambda_c \quad (2.56)$$

The idea is then to replace  $T_0$  in Equation (2.54) by:

$$T_{0a} = T_s - \phi \lambda_c / m c_c \quad (2.57)$$

In the unsteady regime,  $T_{0a}$  is different from  $T_0$ . It is usually called the “apparent” or “effective” initial temperature, e.g. the initial temperature that would correspond to a hypothetical steady-state for the given values of  $T_s$  and  $\phi$ . We can now formulate the previous steady-state laws (2.54) as:

$$m = \bar{m}(\phi, P) \quad (2.58)$$

$$T_s = \bar{T}_s(\phi, P) \quad (2.59)$$

The ZN approach assumes that these laws are valid in the unsteady regime. This is usually accepted, as long as the apparent initial temperature remains within acceptable bounds. It is also required that data for this initial temperature be available, or at least reasonably extrapolated. Note that

this formulation means that the degradation processes are assumed to depend only on the surface temperature and temperature gradient, hence they also behave in a quasi-steady manner. The only unsteady phenomenon is the evolution of the thermal profile within the solid phase.

At time  $t$ , for a given temperature profile in the solid phase  $T(x, t)$ , the heat flux that enters the solid phase is obtained (iteratively) by inverting Equation (2.59): from the known value of  $T_s$ , we find the required value of  $\phi$  such that  $T_s = \bar{T}_s(\phi, P)$ . At each iteration, Equation (2.58) directly gives the surface regression speed.

The ZN approach is therefore relatively straightforward to implement. However, it is mostly dedicated to the study of unsteady combustion around steady state. In particular, it cannot be applied confidently to the study of ignition, because the experimental steady-state laws cannot be constructed for a sufficiently wide range of  $T_s$  values. Ignition studies employing the ZN approach use a temperature criterion to detect ignition [55, 56]. Before ignition, the solid is assumed inert with  $m = 0$ . Upon reaching the user-specified ignition temperature, the gas flame instantaneously appears via the activation of the ZN model. Therefore the transition to ignition is parametrised and approximate. We prefer to use a truly dynamic model that is applicable to all the phases of the ignition transient. Consequently, we do not consider this approach further, except shortly in Section 6.3.1 to assess the stability of various solid propellant configurations. Note that the ZN method can also be employed with steady-state dependencies obtained from steady-state combustion models like the ones mentioned in the introduction of this chapter. For instance, Weber [31] uses the ZN approach and the steady-state relations from the WSB model [62] to simulate the one-dimensional ignition of solid propellants by a laser heat flux. Replacing the experimental steady-state laws by laws obtained from a flame modelling is in essence equivalent to solving the gas phase conservation equations (with the temporal derivatives removed), or using a map of the steady-state gas phase heat flux generated in a similar manner, hence we do not consider that approach further.

## Partial conclusion

In this chapter, we have presented the various modelling choices for the one-dimensional combustion of a solid propellant. The unsteady model that has been derived will be used as a basis model in the rest of this work.

Specificities of the one-dimensional approach for heterogeneous propellants have been mentioned. The treatment of the in-depth and surface degradations has been reviewed and our corresponding modelling choice has been justified. Overall, all existing solid degradation models suffer from limitations, be it in terms of achievable behaviour or available experimental data to fit their parameters. A comprehensive and generic model is yet to be found. The numerical tools developed in the scope of this work consider the classical pyrolysis law for the surface degradation, which is the simplest approach and is routinely used in the literature. The tools can however be trivially adapted to use the AEA approach or other surface pyrolysis laws. The use of the in-depth unsteady degradation model is also possible with some minor changes, such as including the in-depth source term and propellant transformation to solid state pyrolysis products. It is believed that the generic flame model will be able to represent a wide variety of propellant flames decently. The ZN approach, whose use is wide-spread for the study of unsteady combustion dynamics, has been shortly presented, and it has been shown that it cannot be applied to the study of ignition without further simplifications, e.g. use of an ignition temperature criterion.

Overall, the unsteady model we have presented is a model that is more complex than traditional ignition models for large-scale simulations, while remaining slightly simpler than models from state-

of-the-art one-dimensional codes. This intermediate level of refinement offers a great flexibility in the modelling, as will be evident with the various simulations presented in Chapter 6, and is well suited for the scope of the present work.

Some insight into the model's behaviour can be obtained by studying a simplified version of the previous model in steady-state, in particular to obtain a better understanding of the impact of the surface connection conditions on the solution profile. This is the purpose of the next chapter.



# Chapter 3

## Semi-analytical solution in steady-state

### Summary

In this chapter, we introduce a simplified version of the previous detailed model, with emphasis on the steady-state setting. The goal is to enable a mathematical analysis of the structure of the model, and to build a steady-state solution tool that can solve the 1D combustion problem with fewer assumptions than the major 1D analytical models from the literature. Mathematical analysis of the continuous system leads to the demonstration of existence and uniqueness of the temperature profile and of the regression speed, which is the eigenvalue of the problem. The mathematical reasoning directly leads to the construction of a numerical method to obtain the corresponding solution profiles. The associated software will be of great help in the rest of the manuscript by enabling efficient parametric studies to investigate solution sensitivities, and to construct reference solutions to verify other propellant combustion simulation tools.

The generic one-dimensional model presented in the last chapter is detailed and requires advanced space and time discretisation methods to be simulated. Historically, to cope with the inability of simulating such a complex model, many simpler analytical models have been developed. The main ones are the Denison-Baum-Williams (DBW) model [65, 190], the Beckstead-Derr-Price (BDP) model [191], the Ward-Son-Brewster (WSB) model [62, 63], and the KTSS model [192]. These use several assumptions to greatly simplify the equations to be solved. The first assumption is that of steady-state. All time dependences are dropped, and only the steady-state combustion is considered. As in the previous generic model, the pyrolysis remains concentrated at the surface. The thermal properties of the solid phase are assumed constant, which leads to a simple temperature profile solution of exponential form in this phase. The gas phase equations are not analytically solvable, in particular due to strongly nonlinear source term. To circumvent this issue, the analytical models further assume that the gas phase only contains two species: one reactant resulting from the pyrolysis, and one product. There is only one global reaction which transforms the reactant into the product. The Lewis number that controls the species diffusion is assumed unitary, such that species and heat diffuse alike. Even with these simplifications, the chemical source term still leads to an unmanageable analytical resolution. Consequently, all these models have to model the heat-release distribution. For the flame-sheet models DBW and BDP, the activation energy  $E_a$  of the gas phase reaction is very high, hence the chemically-controlled flame results in a heat-release rate that is a Dirac function of space, i.e. heat release only occurs at single point in space, located

at the flame stand-off distance. This allows for the splitting of the gas phase into two separate zones: the convection-diffusion zone and the reaction-diffusion zone, starting at the flame stand-off distance  $x_f$  (model-specific). The equations can be solved analytically in each zone separately and linked at the interface between the two, yielding an explicit or implicit expression for the burning mass flow rate  $m$ . Alternatively, the KTSS model instead assumes a constant heat-release rate between the surface and the flame stand-off distance, which is sensible for diffusion-controlled flames. Note that Culick gives a review of various approaches to model the heat release distribution in [156]. On the opposite, the more recent WSB model assumes that  $E_a$  is zero, which is *a posteriori* shown to lead to better agreement with experimental results for certain classes of solid propellants. This allows for an analytical thermal profile to be derived in the gas phase, as the typical dependence of the reaction rate through an exponential term is removed. In any case, all these models enforce a pressure dependence of the reaction rate, either explicitly or via the concentration exponent  $n_r$  for the reactant. It is found that the obtained steady-state burning rate law is  $aP^n$  with  $n = n_r/2$ .

These models give relations between the propellant physical characteristics and the physical state of the propellant and gas flow (surface temperature, regression speed). They allow for a global understanding of the phenomenon. In all these models, the equations describing the physics of both phases can only be solved for a unique value of the regression speed  $r$ , also called the eigenvalue of the problem. The name eigenvalue is adopted here for two reasons. First, it is historically used in the papers on solid propellant theory as well as in the laminar flame theory. Second, we will show in this chapter that the steady-state problem is a nonlinear eigenvalue problem for an elliptic operator with a nonlinear source term as well as a nonlinear dependency of the regression rate on the surface temperature, and thus a specific case of a general nonlinear eigenvalue problem on the whole real line. It bears some similarity with the eigenvalue problem of a second order linear elliptic operator such as the Laplace operator on a compact interval with proper boundary conditions [193]: we look for both an eigenfunction of space (the thermal profile) and the related real eigenvalue of the operator (the regression speed). Its extension to the nonlinear case has also been studied in the literature [194].

There are several points of interest in studying a steady-state model. First, the strong computational expense of the more detailed models is spared, making large parametric studies very accessible. Second, its simplified equations allow for a better understanding of the interactions between the various phenomena. Finally, the simplified model also allows for an in-depth mathematical analysis of the structure of the problem and of its solution(s). The previously mentioned analytical models rely on strong assumptions, especially regarding the gas phase flame structure. Direct comparison with a more detailed modelling can therefore only be approximate. It is interesting to design an easy-to-use and yet precise analytical or semi-analytical model, which will not require as many assumptions as the existing analytical models, in particular for the gas phase reaction activation energy, thus remaining closer to the physics. The model should be amenable to a full theoretical study of existence and uniqueness, setting the mathematical basis of the model, and its resolution should be simple and efficient, making use of a specific numerical method.

Such a model already exists for travelling combustion waves in laminar premixed flames and has been studied for quite some time, for example in [64]. In this reference, a phase-plane representation of a simplified combustion problem with unitary Lewis number, two species and a single reaction is considered, and the theory of dynamical system leads to the proof of existence and uniqueness of a travelling wave profile, for any value of the gas phase reaction activation energy. The combustion wave speed is shown to be a key parameter for which only one value allows the simplified problem to be solved. This particular value can be determined numerically through a shooting method, for

any value of the activation energy of the gas phase reaction.

In the case of a solid propellant, Verri [195] presented a demonstration of the uniqueness of the travelling wave solution and its stability using a different approach, without modelling the gas phase, but only considering the gas heat feedback as a function of the regression speed with specific mathematical properties. This means the gas heat feedback was assumed to have a unique value for a given regression speed, which tacitly means that the gas phase temperature profile was also assumed unique, although this was not investigated in his paper. His framework was also slightly restrictive, as only exothermic surface reactions were considered, or weakly endothermic ones. No numerical method was developed to determine the solution profile and regression speed. Johnson and Nachbar [196] have also analysed the mathematical behaviour and uniqueness of the eigenvalue for the burning of a monopropellant, but they assumed that the surface temperature is a given constant. They showed that for any reasonable value of this temperature, a single regression speed exists such that the complete problem is solved. Note that other approaches, for example based on the topological degree theory have been used to establish the existence and uniqueness of the travelling wave solution for laminar flames [197, 198] and may be applicable to the solid propellant case with relatively few assumptions. However they do not allow for the derivation of a numerical method to generate solution profiles, which is one of our goals.

In the present chapter, we aim at proving the existence and uniqueness of the solution for a variable surface temperature determined from the regression speed via a pyrolysis law, with non-trivial coupling conditions at the interface, with proper representation of the gas phase, in particular without simplifications regarding the reaction rate. Therefore, we propose a generalisation of the phase-plane approach from Zeldovich to investigate a steady-state model of solid propellant combustion. The model is derived by gradually simplifying the generic detailed model from Chapter 2 that describes the evolution of the temperature of the solid propellant, the evolution of the gas phase, and the pyrolysis of the propellant. The model takes into account thermal expansion and density changes in the gas phase. The nonlinear reaction rate of the single global gas phase reaction is treated without any simplification, as opposed to the previously mentioned analytical models. Compared to the original laminar flame framework of Zeldovich, greater complexity arise from the presence of the solid phase and the constraints related to the surface connection conditions and surface pyrolysis with variable surface temperature. We overcome these additional difficulties through a detailed dynamical system study of the associated heteroclinic orbit in phase space. We study the existence and uniqueness of a travelling wave solution of this system, that is we look for a temperature profile and a wave velocity  $r$ , the so-called eigenvalue or regression velocity. Interpretation of the behaviour of the system in phase space brings a better understanding of the role of the interface and the influence of the different parameters.

Overall, this chapter offers several contributions. First, the new steady-state model relaxes the assumption on activation energy typical of existing analytical models, which consider its value to be either zero or very large. Second, the existence and uniqueness of the steady-state solution is proved, setting a solid mathematical basis for the model. Third, a numerical method is directly derived for the solution of the steady-state problem. Indeed, the phase space approach naturally leads to the development of a very efficient shooting method to iteratively find the speed of the wave and ultimately its profile with arbitrary precision. Fourth, we show at the end of this chapter how the modelling can be gradually complexified to include nonlinear effects, e.g surface radiation.

The content of this chapter is mostly adapted from our article [66] and from the discussion we had with the reviewers. The model presented here cannot be used for the study of ignition, however the numerical method presented here for the steady-state solution will be a very useful verification

tool for the numerical codes presented in the other chapters. Indeed, the simplifications (unitary Lewis, single reaction...) used to derive this steady-state model are all easily reproducible in a more generic model (such as the one from Chapter 2) solved with a CFD approach, which enables a direct comparison of the steady-state solution profiles obtained with each tool.

## 3.1 Simplification of the generic model

In the previous chapter, the generic model accounts for various phenomena, such as unsteady effect, chemical kinetics of arbitrary complexity and others. In the present section, we will gradually introduce simplifying assumptions to derive a simpler model, amenable to the mathematical analysis discussed in the introduction of this chapter.

### 3.1.1 Physical assumptions

We introduce the following set of additional assumptions, also shared by the classical analytical models. First, we assume that the thermal properties of the gas and solid phases are constants:

**H 6.** The specific heat of the solid  $c_c$  is constant. The solid phase thermal conductivity  $\lambda_c$  is constant. The species specific heats  $c_{p,\mathfrak{k}}$  are all equal and constant:  $c_{p,\mathfrak{k}} = c_p \forall \mathfrak{k}$ , with  $c_p$  the constant gas specific heat.

We use further assumptions to simplify the gas phase composition, as well as the diffusion and reaction processes:

**H 7.** The gas phase contains two species: the reactant  $G^1$  and the product  $G^2$ , with mass fractions  $Y_1$  and  $Y_2$ . There is only one irreversible reaction  $\nu_1' G_{(g)}^1 \rightarrow \nu_2'' G_{(g)}^2$ , whose reaction rate  $\tau$  is positive or 0. The species  $G^1$  is completely consumed at  $x = +\infty$ . Both species have the same molar mass  $\mathcal{M}$  and therefore opposite global stoichiometric coefficients. No binary species diffusion takes place in the gas phase, i.e. the non-diagonal components  $D_{ij}$  of the species diffusion matrix are zero for  $k \neq j$ . The species diffusion coefficients  $D_{ii}$  are equal,  $D_{ii} = D_g \forall i$ .

We introduce  $D_{th} = \lambda_g/(\rho c_p)$  the thermal diffusivity of the gas,  $D_c = \lambda_c/(\rho_c c_c)$  the thermal diffusivity of the solid propellant. In the gas phase, we assume that heat and species diffusions are equivalent:

**H 8.** The Lewis number  $Le = D_{th}/D_g$  is 1 in the gas phase.

We simplify our notations by introducing  $D = D_g = D_{th}$ .

Let us now focus on the pyrolysis process. We introduce the solid phase species  $P$  which represent the solid propellant. Having only one reactant in the gas phase, we assume the following:

**H 9.** The pyrolysis reaction transforms the solid propellant  $P$  into the species  $G^1$ .

The complete pyrolysis reaction  $P_{(s)} \rightarrow G_{(g)}^1$  can be decomposed into two successive reactions:

- $P_{(s)} \rightarrow G_{(s)}^1$ , the transformation of the solid propellant  $P_{(s)}$  into the pyrolysis product at solid state  $G_{(s)}^1$ , with the heat of reaction  $Q_s = h_{P_{(s)}}^0 - h_{G_{(s)}^1}^0 + (c_c - c_{P_{G_{(s)}^1}})(T_s - T_{std})$ , with  $c_{P_{G_{(s)}^1}}$  the specific heat of the pyrolysis product  $G^1$  at solid state.

- $G_{(s)}^1 \rightarrow G_{(g)}^1$ , the sublimation of the solid pyrolysis product  $G_{(s)}^1$  into  $G_{(g)}^1$ , at constant temperature  $T_s$ , with the latent heat  $L_v = h_{G_{(g)}^1}(T_s) - h_{G_{(s)}^1}(T_s) = (h_{G_{(g)}^1}^0 - h_{G_{(s)}^1}^0) + (c_p - c_{P_{G_{(s)}^1}})(T_s - T_{std})$

Overall, the heat of the pyrolysis reaction is:

$$Q_p = Q_s - L_v = h_{P_{(s)}}^0 - h_{G_{(g)}^1}^0 + (c_c - c_p)(T_s - T_{std}) \quad (3.1)$$

which depends linearly on  $T_s$ . On the contrary, the heat of the gas phase reaction  $Q$  does not depend on temperature as both species have the same specific heat. The dependence of  $Q_p$  on  $T_s$  leads to a more complex mathematical treatment in this chapter. Therefore, we remove this dependency by imposing the following assumption:

**H 10.** The specific heats of the solid and gas phases are equal:  $c_p = c_c$ .

Even if questionable, Assumption H10 is often used in the literature [19, 63, 120, 199] and the results obtained are still quantitatively correct. Note that the numerical method derived at the end of this chapter is able to handle the case  $c_p \neq c_c$ .

We wish to study steady-state combustion, that is we look for self-similar combustion waves in the Galilean reference frame  $\mathcal{R}_G$ . Following the change of variable used to track the interface in the previous chapter, this is equivalent to the following assumption:

**H 11.** All time derivatives are set to 0.

Finally, potential nonlinearities arising from radiative fluxes are discarded:

**H 12.** Radiative effects are neglected.

This means that no external flux, e.g. laser flux, no radiation from the gas phase and no radiative heat loss from the solid are considered. Radiative heat losses were shown to allow for two different travelling wave solutions to be found at a given pressure [200], with only one being stable. The inclusion of such phenomena in the modelling and its impact on the results obtained in the present investigation are discussed at the end of this chapter.

### 3.1.2 Mathematical assumptions

To allow for an accessible mathematical treatment of the steady-state problem later on in this chapter, we introduce some final assumptions, which do not impact the physics. First we require a certain regularity of the solution profiles.

**H 13.** All solution components are smooth functions of  $x$  in each separate phases, but may be discontinuous at the interface.

Extra properties are also required for the pyrolysis law:

**H 14.** The mass flow rate is  $\mathcal{C}^1$  with respect to  $T_s$  and satisfies the property  $\partial_{T_s} m > 0$ , i.e. the mass flow rate increases with increasing surface temperature.

An additional assumption is also proposed to cope with an issue similar to the cold boundary difficulty resolution [201–203].

**H 15.** The propellant is not be consumed at  $T_s \leq T_0$ ,  $T_0$  being in all realistic cases close to the ambient temperature. That behaviour is usually not depicted exactly by the pyrolysis laws found in the literature, which usually only tend to 0 as  $T_s$  tends to 0. Therefore we introduce a slightly modified pyrolysis law that contains a cut-off so that  $m$  smoothly goes to 0 as  $T_s$  approaches  $T_0$ :

$$m = f(T_s, P) \phi(T_s - T_0) \quad (3.2)$$

with  $\phi$  a smooth function such that  $\phi(y) = 0$  for  $y \leq 0$  and  $\phi(y)$  quickly reaches a value of 1 as  $y$  becomes greater than 0. The function  $\phi$  can typically be a sigmoid function.

The smooth cut-off introduced here is solely used in our theoretical analysis, to facilitate the proof of existence and uniqueness of the travelling wave solution. The numerical shooting method presented in Section 3.3 does not require such a cut-off.

### 3.1.3 Simplified model

Using the previous assumptions, we can simplify the generic detailed model from Chapter 2.

Having only two species, we replace the transport equation for  $Y_2$  by the global mass conservation  $Y_2 = 1 - Y_1$ , and we introduce the stoichiometric coefficient  $\nu = \nu_1 = -\nu_2 = -\nu'_1 < 0$ , and we introduce  $Y = Y_1$  for the sake of simplicity.

In our study  $T_0$  and  $P$  are constants, therefore the mass flow rate will be considered a function of  $T_s$  only for clarity.

We can transform all our partial differential equations and obtain the following system.

#### The simplified travelling wave model

For  $x < 0$ :

$$r d_x T - D_s d_{xx} T = 0 \quad (3.3)$$

For  $x > 0$ :

$$\begin{cases} r d_x \rho + d_x(\rho u) = 0 & (3.4) \\ \rho(u + r) d_x Y - d_x(\rho D_g d_x Y) = \nu \mathcal{M} \hat{\tau} & (3.5) \\ \rho(u + r) d_x T - d_x(\rho D_{th} d_x T) = \frac{\hat{\tau} Q_{mol}}{c_p} & (3.6) \end{cases}$$

with the ideal gas law:

$$\rho = \frac{P \mathcal{M}}{RT} \quad (3.7)$$

The boundary conditions are:

$$\begin{cases} T(-\infty) = T_0 & (3.8) \end{cases}$$

$$\begin{cases} T(0^-) = T(0^+) = T_s & (3.9) \end{cases}$$

$$\begin{cases} (\lambda_s d_x T)_{0^-} = m Q_p + (\lambda_g d_x T)_{0^+} & (3.10) \end{cases}$$

$$\begin{cases} (m Y)_{0^-} = (m Y - \rho D_g d_x Y)_{0^+} & (3.11) \end{cases}$$

$$\begin{cases} d_x T(+\infty) = 0 & (3.12) \end{cases}$$

$$\begin{cases} d_x Y(+\infty) = 0 & (3.13) \end{cases}$$

As before, the pyrolysis mass flow rate  $m$  is given by the pyrolysis law (3.2).

**Key steps to obtain the set of simplified equations** In the full system, the first two terms of all gas phase transport equations can be simplified by using the continuity equation, for example:

$$\partial_t \rho Y_i + \partial_x \rho u Y_i = \rho \partial_t Y_i + \rho u \partial_x Y_i$$

In the gas phase energy equation (2.26), we can expand the term of heat diffusion caused by chemical diffusion, using our assumptions that all species specific heats are equal:  $\rho \partial_x T \sum_{i=1}^{n_e} c_{p,i} Y_i V_i = \rho c_p \partial_x T \sum_{i=1}^{n_e} Y_i V_i = 0$ , by definition of the diffusion velocities.

The term of heat production due to the single chemical reaction can be simplified as follows:

$$-\sum_{i=1}^{n_e} h_i \omega_i = -\sum_{i=1}^{n_e} \omega_i (h_i^0 + c_p(T - T_{std})) = -\tau \sum_{i=1}^{n_e} \nu_i \mathcal{M}_i (h_i^0 + c_p(T - T_{std}))$$

Using our assumption that all species molar masses are equal, the term in  $c_p$  disappears according to mass conservation ( $\sum_{i=1}^{n_e} \nu_i \mathcal{M}_i = 0$ ):

$$-\tau \sum_{i=1}^{n_e} \nu_i \mathcal{M}_i (h_i^0 + c_p(T - T_{std})) = -\tau \mathcal{M} \sum_{i=1}^{n_e} \nu_i h_i^0 = \tau Q_{mol}$$

### 3.1.4 Conservation properties

Starting from a detailed modelling of the different processes at stake, we have introduced gradual simplifications based on several physical assumptions. We may now perform simple mathematical manipulations on the travelling wave model to derive several balance equations. These considerations will allow us to obtain characteristic scales from which dimensionless variables can be formed.

**Proposition 3.1.1.** *The conservation of the mass flow rate implies for  $x > 0$ :*

$$\rho(x)(u(x) + r) = \rho_s r = m$$

*Proof.* We integrate the continuity equation (3.4) in the gas phase from  $0^+$  to  $x$ . We obtain:

$$\rho(x)(u(x) + r) = \rho(0^+)(u(0^+) + r)$$

Following Assumption H3, no accumulation takes places at the interface. Hence the gas mass flow rate must be equal to the rate of propellant mass consumption  $-\rho_s r$ , which is the proposed result.  $\square$

**Remark 1.** *We explained in Chapter 2 the reasons why we do not consider the momentum equation in the general unsteady model. In the low-Mach framework, this equation would, in steady-state, essentially reduce to:  $m d_x u = d_x \tilde{P}$ , with  $\tilde{P}$  the hydrodynamic pressure, which is a pressure perturbation of the order of  $\text{Ma}^2$ , with  $\text{Ma}$  the Mach number. Due to the one-dimensionality of our approach, the velocity field is directly related to the spatial evolution of  $\rho$  through the continuity equation. Hence, the momentum equation is not needed to determine  $u$ . We may still use it to determine the hydrodynamic pressure field. In particular, we would find that the hydrodynamic pressure is increasing with  $x$ , and that the overall pressure variation across the gas phase is  $\Delta P = -m \Delta u$ . Typically we*

obtain  $-10$  Pa, which is considerably lower than the average pressure (around 1 to 10 MPa). This legitimates our assumption of uniform  $P$ . Had we used a multidimensional approach, we would not have been able to decouple the velocity field from the hydrodynamic pressure field, and we would have needed to include the momentum equation.

**Proposition 3.1.2.** *The complete consumption of  $G^1$  implies:*

$$\int_0^{+\infty} \tau(T(x), Y(x)) dx = -\frac{m}{\mathcal{M}\nu}$$

*Proof.* We integrate the species transport equation (3.5) in the gas phase from  $0^+$  to  $+\infty$ , utilizing Proposition 3.1.1. As all gradients are zero at  $+\infty$ , we get:

$$-mY(0^+) + \rho(0^+)Dd_xY(0^+) = \mathcal{M}\nu \int_{0^+}^{+\infty} \tau(T(x), Y(x)) dx$$

Using Equation (3.11), the left-hand side is equal to  $-mY(0^-)$ . Following H9, we have  $Y(0^-) = 1$ , therefore we obtain the proposed result.  $\square$

We introduce  $Q = -Q_{mol}/(\nu\mathcal{M})$  the heat of reaction in the gas phase per unit mass of  $G^1_{(g)}$  consumed. As  $\nu < 0$ ,  $Q$  and  $Q_{mol}$  are both positive.

**Proposition 3.1.3.** *The burnt gas temperature at  $x = +\infty$  is  $T_f = T_0 + (Q + Q_p)/c_p$ .*

*Proof.* Integrating the energy equation (3.6) in the gas phase between 0 and  $+\infty$  and using Proposition 3.1.2, we obtain:

$$m(T_f - T_s) = -\frac{\lambda_g}{c_p}d_xT(0^+) + \frac{Q_{mol}}{c_p} \int_0^{+\infty} \tau(T(x), Y(x)) dx = -\frac{\lambda_g}{c_p}d_xT(0^+) + \frac{mQ}{c_p}$$

Integrating the heat equation in the solid phase (3.3) between  $-\infty$  and 0 and using Proposition 3.1.1, we can write:

$$m(T_s - T_0) = +\frac{\lambda_s}{c_c}d_xT(0^-)$$

Using the interface boundary condition (3.9) and Assumption H10, we can combine both energy balances and obtain the proposed result.  $\square$

**Remark 2.** *If we do not assume  $c_c = c_p$ , the balance reads:*

$$T_f = \left(1 - \frac{c_c}{c_p}\right) T_s + \frac{c_c}{c_p} T_0 + \frac{Q + Q_p}{c_p}$$

*This formula appears in many papers, however the dependence in  $T_s$  is fictitious and may be misleading [183]. Indeed it is expected that the complete energy balance does not depend on the mass flow rate. The dependence is removed when using the standard enthalpies to express  $Q_p$  as a function of  $T_s$  (see Section 3.1.1):*

$$T_f = \frac{c_c}{c_p} T_0 + \frac{h_{P(s)}^0 - h_{G^2(g)}^0}{c_p} + \left(1 - \frac{c_c}{c_p}\right) T_{std}$$

*This expression can also be directly obtained by performing a simple energy balance between  $-\infty$  and  $+\infty$ , neglecting the kinetic energy.*

**Proposition 3.1.4.** *Under Assumption H8, we can define a combustion enthalpy which is constant in the gas phase:*

$$h = -\frac{Q_{mol}}{\nu\mathcal{M}}Y + c_p(T - T_0) = c_p(T_f - T_0)$$

*Proof.* We introduce  $\check{Y} = YQ_{mol}/(\nu\mathcal{M})$  and  $\check{T} = c_p(T - T_0)$ . We convert Equations (3.5) and (3.6) to our new variables. Using Assumption H8, introducing  $\beta = \lambda_g/c_p = \rho D$  which is a constant and  $h = \check{T} - \check{Y}$ , we obtain:

$$md_x h = \beta d_{xx} h$$

We can integrate this expression between the interface  $0^+$  and  $x$ :

$$h(x) = h(0^+) + d_x h(0^+) \int_0^x \exp\left(\frac{my}{\beta}\right) dy$$

Using the boundary conditions at  $+\infty$ , we get  $d_x h(+\infty) = 0$ . Alternatively,  $Y$  and  $T$  being bounded,  $h$  is bounded too. Both considerations lead to  $d_x h(0^+) = 0$ , hence we get:  $h(x) = h(+\infty) = c_p(T_f - T_0)$ .  $\square$

### 3.1.5 Dimensionless equations

Using the equations from the simplified travelling wave model and the results obtained in the previous subsection, we can now write dimensionless equations for our problem. We introduce  $L$  a constant length scale. For the solid phase, we take Equation (3.3), divide it by  $(T_f - T_0)$  to let  $\theta$  appear, and switch the spatial derivatives from  $x$  to  $\tilde{x}$  ( $Ld_x = d_{\tilde{x}}$ ); we then multiply it by  $L^2/D_s$  and use the definition of  $\gamma$  to obtain a dimensionless equation. For the gas phase, Proposition 3.1.4 allows us to express  $Y$  as a function of  $T$ . Therefore the reaction rate  $\tau(T, Y)$  can be expressed as a function of temperature only  $\tilde{\tau}(\tilde{\theta})$ . A dimensionless equation can then be derived in a similar fashion as for the solid phase. Eventually, a first non-dimensionalisation of the interface thermal balance condition leads to the construction of a term  $\tilde{S}(\tilde{r}) = -rL\rho_s Q_p(T_s)/(\lambda_g(T_f - T_0)) = -\tilde{r}\rho_s D_s Q_p(T_s)/(\lambda_g(T_f - T_0))$ . Recalling the definitions of  $\eta$ ,  $D_s$ , we obtain:

$$\tilde{S}(\tilde{r}) = -\frac{\eta Q_p(T_s)}{c_c(T_f - T_0)} \tilde{r} \quad (3.14)$$

where the denominator can be further simplified by using Assumption H10 and the global energy balance from Proposition 3.1.3 Dimensionless equivalents of the continuity of the temperature across the surface and the boundary conditions at infinity are trivially derived from the simplified travelling wave model. Overall, we can construct the following system.

**Dimensionless system** Introducing  $\tilde{x} = \frac{x}{L}$ ,  $\tilde{r} = \frac{rL}{D_s}$ ,  $\tilde{\theta} = \frac{T(\tilde{x}) - T_0}{T_f - T_0}$ ,  $\eta = \frac{\lambda_s}{\lambda_g}$ , using the notation  $\cdot' = d_{\tilde{x}} \cdot$  and  $\tilde{\gamma}(\tilde{x}) = \tilde{\theta}'(\tilde{x})$  we have:

$$\begin{cases} \tilde{\theta}' = \tilde{\gamma} & (3.15) \\ -\tilde{r}\tilde{\gamma} + \tilde{\gamma}' = 0 & \text{for } \tilde{x} < 0 & (3.16) \\ -\eta\tilde{r}\tilde{\gamma} + \tilde{\gamma}' = -\tilde{\Psi} & \text{for } \tilde{x} > 0 & (3.17) \end{cases}$$

with the dimensionless heat source term:

$$\tilde{\Psi}(\tilde{x}) = \frac{L^2 Q_{mol}}{\lambda_g (T_f - T_0)} \tilde{\tau}(\tilde{\theta}(\tilde{x})) \geq 0 \quad (3.18)$$

The associated boundary conditions are:

$$\tilde{\theta}(-\infty) = 0 \quad (3.19)$$

$$\tilde{\theta}(0^-) = \tilde{\theta}(0^+) = \tilde{\theta}_s(\tilde{r}) \quad (3.20)$$

$$\tilde{\gamma}(0^+) - \eta \tilde{\gamma}(0^-) = \tilde{S}(\tilde{r}) \quad (3.21)$$

$$\tilde{\theta}(+\infty) = 1 \quad (3.22)$$

$$\tilde{\gamma}(-\infty) = 0 \quad (3.23)$$

$$\tilde{\gamma}(+\infty) = 0 \quad (3.24)$$

with the target interface balance:

$$\tilde{S}(\tilde{r}) = -\eta \frac{Q_p}{Q_p + Q} \tilde{r} \quad (3.25)$$

**Remark 3.** For a given value of  $\tilde{r}$ ,  $\tilde{\theta}_s$  is given by (3.2). Therefore the first-order ODEs (3.15) and (3.16) can be integrated from  $\tilde{x} = -\infty$  to 0, using the boundary conditions (3.19) and (3.20), and the solution profiles for  $\tilde{\theta}$  and  $\tilde{\gamma}$  are unique. Similarly the first-order ODEs (3.15) and (3.17) may also be integrated from  $\tilde{x} = +\infty$  to 0, using the boundary conditions (3.22) and (3.20), and the solution profiles are also unique. Boundary conditions (3.23) and (3.24) are only introduced to emphasise the behaviour of the system at infinity, however they are not mathematically required. The difficulty arises from the interface thermal balance (3.21) which overconstrains our system. For a random value of  $\tilde{r}$ , it is likely that this condition will not be satisfied. However the dependence of this condition on  $\tilde{r}$  through  $\gamma$  and the target interface balance  $\tilde{S}$  allows us to envision that some specific values of  $\tilde{r}$  might lead to this condition being verified (hence the name “target” for  $\tilde{S}$ ). Therefore, the dimensionless regression velocity  $\tilde{r}$  is a key variable and can be considered as an “eigenvalue” of the problem.

**Remark 4.** The dimensionless heat source term  $\tilde{\Psi}$  has the same behaviour as the reaction rate  $\tilde{\tau}$ . It is positive for  $\tilde{\theta} \in [0, 1]$  and vanishes for  $\tilde{\theta} = 1$ , since all the fuel is burned, i.e.  $\tilde{\Psi}(1) = 0$ .

**Remark 5.** The sign of the temperature gradient jump across the interface  $[d_x T]_{0^-}^{0^+}$ , or equivalently  $[d_x \tilde{\theta}]_{0^-}^{0^+}$ , depends on three factors:

- $\eta = \lambda_s / \lambda_g$  the ratio of the thermal conductivities in the gas and in the solid
- $Q_p$  the reaction heat of the pyrolysis reaction (see Section 3.1.1)
- $T_s$ , which is directly related to the regression rate

Let us underline that the presence of the ratio  $\eta$  of thermal conductivities may have a strong impact on the sign of the jump. As an example, in a configuration where  $Q_p = 0$ , we have  $\tilde{S}(\tilde{r}) = 0$ . If  $\eta > 1$ , then  $d_x T(0^+) > d_x T(0^-)$ , but if  $\eta < 1$ , then  $d_x T(0^+) < d_x T(0^-)$ .

**Remark 6.** With the reactant mass fraction  $Y$  now removed from our set of variables, the species interface condition (3.11) is not considered any more. It can actually be shown that, as long as the thermal interface condition (3.21) is satisfied, this condition is automatically fulfilled, even if  $Le \neq 1$ .

## 3.2 Existence and uniqueness of a travelling wave solution profile and velocity

In this section, we will use the previously established dimensionless system to prove that there exists at least one value of the regression velocity  $r$  such that all boundary conditions can be satisfied and the complete travelling wave problem can be solved. We then proceed to show that there is only one such value of  $r$ . For the sake of simplicity, we drop the “ $\sim$ ” notation.

### 3.2.1 Monotonicity of the temperature profile

A first step in our proof of existence is to show that the temperature profile is increasing in the gas phase.

**Proposition 3.2.1.** *The temperature  $T$  is an increasing function of  $x$  on  $\mathbb{R}$ . There exists a bijection  $g : \mathbb{R} \rightarrow [0, 1]$  such that  $g(x) = \theta$ .*

*Proof.* This proposition is established by considering the behaviour of the temperature in the two phases successively.

**Solid phase** Let  $x_0 \in (-\infty, 0)$  the position of a local extremum for  $\theta$ :  $\gamma(x_0) = 0$ . From Equation (3.16) we get  $\gamma'(x_0) = 0$ . If we integrate Equation (3.15) from  $-\infty$  to  $x_0$ , we get  $\theta(x_0) = 0$ . Therefore no local extremum can be lower than 0. If a local maximum exists at  $x_0$ ,  $\theta(x_0) = 0$ . As  $\theta(-\infty) = 0$ , there would then exist a local minimum  $x_1 < x_0$ , and we must have  $\theta(x_1) < 0$ , which contradicts our previous finding. Therefore no local extremum exists for  $\theta$  in the solid phase. As  $r > 0$  implies  $\theta(0) = \theta_s > 0$  (see Equation (3.2)), we can conclude that  $\theta$  is increasing in this phase.

**Gas phase** We want to prove that the temperature profile is monotonous and increasing in the gas phase. Using a *reductio ad absurdum*, let us suppose that  $\exists x_0 / \gamma(x_0) = 0$ , local extremum or inflection point for  $\theta$ . The energy equation (3.17) then reads:  $\theta''(x_0) = \gamma'(x_0) = -\Psi(\theta(x_0)) < 0$ , which means that  $x_0$  can only be a local maximum. Consequently there exists a local minimum at  $x_1 \in (x_0, +\infty)$  such that  $\gamma(x_1) = 0$  and  $\gamma(x) < 0 \forall x \in (x_0, x_1)$ . We obviously have  $\theta(x_1) < \theta(x_0)$ . Integrating Equation (3.17) from  $x_0$  to  $x_1$  yields:  $-\eta r [\theta(x_1) - \theta(x_0)] = -\int_{x_0}^{x_1} \Psi(\theta(x)) dx$ . The left-hand side is strictly positive, but the right-hand side is strictly negative, consequently there exists no local maximum  $x_0$ . Overall,  $\theta$  does not have any local extremum in the gas phase, and as  $\theta(+\infty) > \theta(0^+)$ , we can conclude that  $\theta$  is monotonous and increasing in the gas phase. This proof is the consequence of a much more general principle in the study of second order elliptic equations called the maximum principle [204].

**Overall Monotonicity** The boundary condition  $\theta(0^+) = \theta(0^-) = \theta_s$  and the requirement  $\theta_s > 0$  allow us to prove that  $\theta$  is increasing and strictly monotonous across both phases. Therefore, we can build a bijection  $g : \mathbb{R} \rightarrow [0, 1]$  such that  $g(x) = \theta$ . This proof is valid even if the regression velocity  $r$  is such that the interface thermal balance (3.21) is not satisfied.  $\square$

We now make use of the monotonicity of  $\theta$  to switch from a spatial point of view to a phase space one. The bijection between  $\theta$  and  $x$  allows for a variable change from  $x$  to  $\theta$  in our equations. Therefore,  $\gamma$  may be considered a function of  $\theta$ . We also have the relation  $d_x \gamma = d_\theta \gamma d_x \theta = \gamma d_\theta \gamma$ . We can transform the dimensionless system into the following one, which we will use to determine the orbit of our system in the phase plane  $(\theta, \gamma)$ .

**Reduced dynamical system for orbit evaluation** The dimensionless system is equivalent to the following set of first-order ODEs and boundary conditions:

$$\begin{cases} -r\gamma(\theta) + \gamma(\theta)d_\theta\gamma(\theta) = 0 & \forall \theta \in [0, \theta_s(r)] \\ -\eta r\gamma(\theta) + \gamma(\theta)d_\theta\gamma(\theta) = -\Psi(\theta) & \forall \theta \in [\theta_s(r), 1] \end{cases} \quad (3.26)$$

$$\begin{cases} \gamma(0) = 0 & (3.28) \\ \gamma(1) = 0 & (3.29) \\ \gamma(\theta_s^+) - \eta\gamma(\theta_s^-) = S(r) & (3.30) \end{cases}$$

**Remark 7.** *This set of equations is similar to the one obtained by Zeldovich et al. [64] for a homogeneous gaseous laminar flame. In this reference, the phase portrait of the temperature profile is also split in two parts. The first one represents the part of the profile where the temperature is lower than an artificial cut-off temperature  $\theta_{\text{ignition}}$ , below which the reaction rate  $\Psi$  is forced to zero. This allows the “cold boundary” problem [201–203] to be overcome. The zone where  $\theta < \theta_{\text{ignition}}$  is purely a convection-diffusion zone. The second part of the laminar flame phase portrait is the same as ours: the gas phase undergoes a reaction which creates a steep increase in temperature before reaching the adiabatic combustion temperature behind the combustion wave. This is a convection-diffusion-reaction zone. The two zones are joined using the continuity of the temperature profile and its gradient, as no reaction or heat accumulation takes place at the interface. In our case, the first part of the phase portrait is not associated with a cut-off of the gas phase reaction rate, but with the fact that the solid phase is inert, therefore it only heats up through thermal diffusion. Our problem thus differs in two ways from the laminar flame one. First, the pyrolysis process is concentrated at the interface and causes a discontinuity of the temperature gradient, which depends on the wave velocity  $r$ . Secondly, the position  $\theta_s$  of the interface in the phase portrait also varies with  $r$ , whereas  $\theta_{\text{ignition}}$  is an arbitrary constant. We can artificially make our problem equivalent to the laminar flame’s one by forcing  $Q_p = 0$ ,  $\eta = 1$ ,  $\theta_s = \theta_{\text{ignition}}$  (no pyrolysis law),  $D_s = D_g$  (and  $c_c = c_p$  as assumed in H10).*

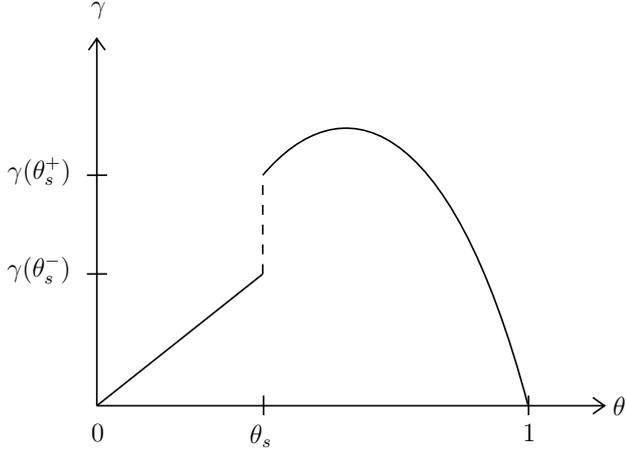
The rest of the study will be based on the analysis of the phase portrait of the system, i.e. the plot of  $\gamma$  versus  $\theta$ . Such a phase portrait is represented in Figure 3.1.

### 3.2.2 Existence of a solution

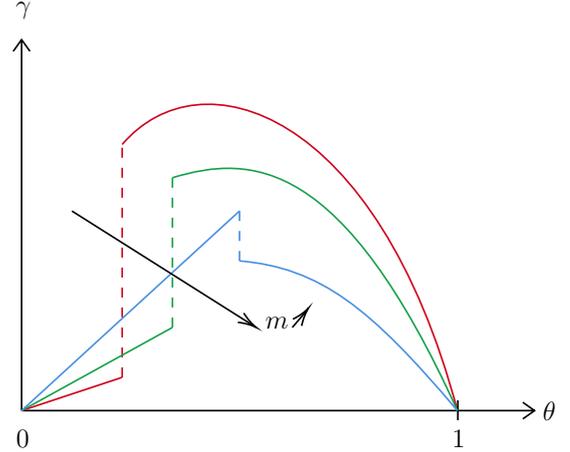
We will now show that there exists at least one wave velocity  $r > 0$  such that the travelling wave problem previously stated has a solution for fixed values of  $P$  and  $T_0$ . We introduce  $\Delta\gamma(r) = \gamma[\theta_s^+(r)] - \eta\gamma[\theta_s^-(r)]$ , the effective interface balance obtained for the regression velocity  $r$  and  $\xi(r) = \Delta\gamma(r) - S(r)$ , which we will call the interface balance mismatch. We introduce another assumption, non-restrictive for any real application:

**H 16.** The heat of the pyrolysis reaction  $Q_p$  is such that  $Q_p > -Q$ .

**Proposition 3.2.2.** *Under Assumptions H15 and H16, there exists at least one wave velocity  $r > 0$  such that the problem stated in the reduced system can be solved. All solutions for the wave velocity  $r$  reside in the interval  $(0, r_{\text{max}})$ , with  $r_{\text{max}}$  the dimensionless wave velocity such that  $\theta_s(r_{\text{max}}) = 1$ .*



**Figure 3.1** Schematic phase portrait in both phases



**Figure 3.2** Evolution of the phase portrait with  $m$ . Each curve corresponds to the solution curve for one value of  $m$ .

*Proof.* The global phase portrait in the gas and solid phases is schematically represented in Figure 3.1. There is a jump of  $\gamma$  at  $\theta_s$ , as explained in Remark 3. More precisely, the thermal boundary condition (3.30) may be reformulated as:

$$\Delta\gamma(r) = S(r) \Leftrightarrow \xi(r) = 0$$

with  $\Delta\gamma$  the effective interface balance, i.e. the dimensionless interface heat fluxes balance we obtain for a given value of  $r$  by integrating Equations (3.26) and (3.27) separately, with boundary conditions (3.28) and (3.29) respectively. The interface balance mismatch  $\xi$  is non-zero when the interface thermal balance condition (3.30) is not satisfied. A complete solution to the reduced system may only be found if there exists a regression velocity  $r$  such that  $\xi(r) = 0$ .

To prove the existence of such a value of  $r$ , we focus on the behaviour of  $\xi$ . We first aim at proving that  $\xi$  is a continuous function of  $r$ . To do so, we add  $r$  as a variable in our reduced system, subject to  $d_\theta r = 0$  with the boundary condition  $r(0) = r_0$  determined from the pyrolysis law (3.2). The reduced system in the solid phase can be recast to the following form:

$$\frac{d}{d\theta} \begin{pmatrix} \gamma \\ r \end{pmatrix} = f \begin{pmatrix} \gamma \\ r \end{pmatrix}, \text{ with } f \begin{pmatrix} \gamma \\ r \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \text{ and initial conditions } \begin{pmatrix} \gamma(0) \\ r(0) \end{pmatrix} = \begin{pmatrix} \gamma_0 \\ r_0 \end{pmatrix} = \begin{pmatrix} 0 \\ m/\rho_c \end{pmatrix}.$$

The associated flow is  $\phi : (\theta; \gamma_0, r_0) \rightarrow (\gamma(\theta), r(\theta))$ . The theory of dynamical systems shows that,  $f$  being here a  $\mathcal{C}^\infty$  function, the flow is also  $\mathcal{C}^\infty$  with respect to the initial conditions. In particular, the solution profile for  $\gamma$  in the solid phase depends continuously on  $r_0 = m/\rho_s$ . As we also assume (Assumption H14) that the surface temperature  $\theta_s$  is a  $\mathcal{C}^\infty$  function of  $r$ ,  $\gamma[\theta_s^-(r)]$  is  $\mathcal{C}^\infty$  with respect to  $r$ . The same reasoning can be applied to the gas phase for  $\gamma[\theta_s^+(r)]$ , so that  $\Delta\gamma$  is  $\mathcal{C}^\infty$ .  $S$  is also trivially a  $\mathcal{C}^\infty$  function of  $r$ . As  $\xi$  is a sum of  $\mathcal{C}^\infty$  functions of  $r$ , we conclude that  $\xi$  is  $\mathcal{C}^\infty$  with respect to  $r$ .

Inspired by this property, we aim at finding two values of the wave velocity  $r_1$  and  $r_2$  such that  $\xi(r_1)$  and  $\xi(r_2)$  have opposite signs, implying that there is at least one value of  $r \in (r_1, r_2)$  such that  $\xi(r) = 0$ . We exhibit two limit cases for the wave velocity  $r$ , which naturally yield a different sign for  $\xi$ :

- **Case  $r = 0$ :** In this case  $m = 0$ , i.e. the solid propellant remains inert. Using Assumption H15, this equates to  $\theta_s = 0$ . The temperature is uniform inside the solid phase. Note that we are still satisfying the monotonicity property of  $\theta$  shown in Proposition 3.2.1 (the monotonicity is strict in the solid phase only if  $\theta_s > 0$ ). With  $r = 0$ , Proposition 3.1.1 yields  $u = 0$ . Equations (3.25), (3.26) and (3.27) lead to:  $\gamma(\theta_s^-(0)) = 0$ ,  $\gamma(\theta_s^+(0)) = (2 \int_0^1 \Psi(y) dy)^{1/2} = (2I_0)^{1/2}$  and  $S(0) = 0$ . Consequently  $\Delta\gamma(0) = (2I_0)^{1/2} > S(0)$ , and therefore  $\xi(0) > 0$ .
- **Case  $r = r_{max}$ :** The solution we are looking for is monotonous and thus requires  $\theta_s \leq 1$ . Based on the pyrolysis law (3.2), the case  $\theta_s = 1$  corresponds to a certain value  $r_{max} < 0$  of the wave velocity. We can then directly integrate the reduced system equations (3.26) and (3.27) to obtain:  $\gamma(\theta_s^-) = r_{max}$ ,  $\gamma(\theta_s^+) = 0$  and  $S(r) = -\eta r_{max} Q_p / (Q_p + Q)$ . Thus,  $\xi(r_{max}) = \Delta\gamma(r_{max}) - S(r_{max}) = -\eta r_{max} (1 - Q_p / (Q_p + Q)) = -\eta r_{max} Q / (Q_p + Q)$ . Assuming H16, we obtain  $\xi(r_{max}) < 0$ .

In realistic cases for  $Q_p$ , we have shown that  $\xi(0) > 0$  and  $\xi(r_{max}) < 0$ . Therefore, as  $\xi$  is a continuous function of  $r$ , there exists at least one value of  $r \in (0, r_{max})$  such that  $\xi(r) = 0$ . Potential solutions with  $r < 0$  or  $r > r_{max}$  are physically meaningless and are not further considered. The existence of a solution for the reduced system implies that a solution also exists for the dimensionless system and the simplified travelling wave system.  $\square$

### 3.2.3 Uniqueness of the solution

Having proved that there exists at least one value of  $r$  such that the travelling wave problem can be solved, we now proceed to show that there is only one such value. There are two cases, depending on the sign of  $Q_p$ .

**Proposition 3.2.3.** *If  $Q_p < 0$ , there exists a unique value of the wave velocity  $r$  such that the reduced system has a solution.*

*Proof.* Studying the existence of a solution in Proposition 3.2.2, we have introduced the interface balance mismatch  $\xi(r) = \Delta\gamma(r) - S(r)$ . A solution to the reduced system with regression velocity  $r$  only exists if  $\xi(r) = 0$ . We have shown that  $\xi$  undergoes a change of sign between  $r = 0$  and  $r = r_{max}$ . This implies that there exists at least one value of  $r$  such that  $\xi(r) = 0$ . As we aim at proving that there is only one such value of  $r$ , we need to show that  $\xi$  is a monotonous function of  $r$ . To do so, we will study separately the evolution of the two terms appearing in the definition of  $\xi$ :  $\Delta\gamma$  and  $S$ . For improved readability, we introduce  $\gamma^- = \gamma(\theta_s^-, r)$  and  $\gamma^+ = \gamma(\theta_s^+, r)$ .

*Evolution of  $\Delta\gamma$*  We have  $\Delta\gamma = \gamma^+ - \eta\gamma^-$ . To study the evolution  $\Delta\gamma$ , we will first analyse the behaviour of  $\gamma^-$  and  $\gamma^+$ .

In the solid phase, we have seen that we may solve Equation (3.26) analytically and find  $\gamma^- = r\theta_s(r)$ . Deriving with respect to  $r$ , we obtain:

$$d_r\gamma^- = \theta_s(r) + r\partial_r\theta_s(r)$$

Assumption H14 on the pyrolysis law (3.2) implies  $d_{T_s}r > 0$  and consequently  $d_r\theta_s > 0$ . Therefore  $\partial_r\gamma^- > 0$ , i.e. that is the more  $r$  increases, the more thermal power is needed to maintain the solid phase temperature profile, as we would expect.

We now focus on the evolution of  $\gamma^+$ . In the gas phase, the dimensionless temperature gradient at the interface is given by integrating Equation (3.27):

$$\gamma^+ = \int_1^{\theta_s(r)} d_\theta \gamma \, d\theta = \int_1^{\theta_s(r)} \left( \frac{-\Psi(\theta)}{\gamma(\theta)} + \eta r \right) d\theta$$

Deriving this expression with respect to  $r$  yields:

$$d_r \gamma^+ = \underbrace{\int_1^{\theta_s(r)} d_r(d_\theta \gamma) d\theta}_{A = \left( \frac{\partial \gamma^+}{\partial r} \right) \Big|_{\theta_s}} + \underbrace{\left( \frac{-\Psi(\theta_s)}{\gamma^+} + \eta r \right) d_r \theta_s}_{B = \left( \frac{\partial \gamma^+}{\partial r} \right) \Big|_r}$$

Let us study the sign of  $A$  and  $B$ . The term  $A$  is the derivative of  $\gamma^+$  with respect to  $r$  at constant  $\theta_s$ . Its sign may be found by following the same reasoning as Zeldovich in his work on laminar flames [64], which we reproduce hereafter for the sake of completeness. Deriving Equation (3.27) with respect to  $r$ , we obtain:  $\partial_r(\partial_\theta \gamma) = \eta + (\partial_r \gamma / \gamma^2) \Psi(\theta)$ . Introducing  $y = -\gamma / \eta$  and  $\Pi(\theta) = \Psi(\theta) / \eta^2$  yields  $\partial_r(\partial_\theta y) = -1 + (\partial_r y / y^2) \Pi(\theta)$ . Zeldovich has shown ([64], page 256) that the solution to this equation is:  $\partial_r y(\theta) = \exp(\chi(\theta)) \int_\theta^1 \exp(-\chi(z)) dz$ , with  $\chi = \int \Pi / y^2$ . Therefore  $d_r y(\theta) > 0$ ,  $\forall \theta \in [0, 1]$  and consequently  $A = -\eta \partial_r y(\theta_s) < 0$ . Physically this means that, for a fixed surface temperature, the heat flux transmitted from the gas phase into the solid phase is decreasing as the mass flow rate increases, which seems logical because the flame tends to be blown away from the surface.

Let us now determine the sign of  $B$ . Based on Assumption H7, we have  $\Psi \geq 0$ . The monotonicity of the temperature profile in the gas phase implies  $\gamma^+ > 0$ , and Assumption H14 leads to  $d_r \theta_s > 0$ . Therefore  $B$  is negative only if  $\Psi(\theta_s) / \gamma^+ > \eta r$ , which may not always be true, thus we cannot directly conclude on the sign of  $d_r \gamma^+$ .

However, if we combine the derivatives of  $\gamma^+$  and  $\gamma^-$  to express the derivative of  $\Delta\gamma$ , the terms containing  $\eta r$  cancel out:

$$\begin{aligned} d_r \Delta\gamma &= d_r \gamma^+ - \eta d_r \gamma^- = A + \left( \frac{-\Psi(\theta_s)}{\gamma^+} + \eta r \right) d_r \theta_s - \eta(\theta_s + r d_r \theta_s) \\ &\Rightarrow d_r \Delta\gamma = A - \eta \theta_s - \frac{\Psi(\theta_s)}{\gamma^+} d_r \theta_s \end{aligned}$$

Overall, the three remaining terms are negative, hence  $d_r \Delta\gamma < 0$ .

*Evolution of  $S$*  Deriving Equation (3.25), we get:  $\partial_r S = -\eta Q_p / (Q_p + Q)$ . Following Assumption H16 ( $Q_p > -Q$ ), we conclude that  $\partial_r S$  has the opposite sign of  $Q_p$ . In this proposition, we assume  $Q_p \in (-Q, 0]$ , therefore we obtain  $\partial_r S > 0$ .

*Evolution of  $\xi$*  We now have determined the signs of each term appearing in the derivative of the interface balance mismatch  $\xi$  with respect to  $r$ . Using the previously established relations, we can write:

$$\partial_r \xi = \partial_r \Delta\gamma - \partial_r S < 0$$

We conclude that  $\xi$  is a monotonous function of  $r$ . We have shown in the proof of Proposition 3.2.2 that  $\xi(r_{max}) < 0$  and  $\xi(0) > 0$ , i.e. that there exists at least one solution wave velocity  $r$  such that the reduced system is solved. The monotonicity of  $\xi$  we just established is the additional property needed to prove that there is only one such solution. Physical interpretations of the behaviour of  $\xi$ ,  $\Delta\gamma$  and  $S$  are given in 3.2.5.  $\square$

**Proposition 3.2.4.** *If  $Q_p > 0$ , there exists a unique value of the wave velocity  $r$  such that the problem stated in the reduced system can be solved. This solution  $r$  belongs to the interval  $(r_{min}, r_{max})$  with  $r_{min}$  such that  $\theta_s(r_{min}) = Q_p/(Q + Q_p)$ .*

*Proof.* This result is obtained in a manner almost identical to the previous one. The difference lies in the behaviour of  $S$ . With  $Q_p > 0$ , we have  $\partial_r S < 0$  and, as  $\partial_r \Delta\gamma < 0$ , we cannot directly conclude on the sign of  $\partial_r \xi$  for  $r \in (0, r_{max})$ . To circumvent this difficulty, we will show that there exists a value  $r_{min}$  such that we always have  $r > r_{min}$ , which verifies  $\xi(r_{min}) > 0$ , and such that  $\xi$  is monotonous on the interval  $(r_{min}, r_{max})$ . Starting from the relations established in the proof of Proposition 3.2.3, we can express  $d_r \xi$ :

$$d_r \xi = d_r \Delta\gamma - d_r S = \underbrace{A - \frac{\Psi(\theta_s)}{\gamma^+} d_r \theta_s}_{<0} + \eta \left( -\theta_s + \frac{Q_p}{Q + Q_p} \right)$$

Consequently, to ensure  $d_r \xi < 0$ , it is sufficient that the last term is negative:

$$\theta_s > \theta_{s,min} = \frac{Q_p}{Q_p + Q} = \frac{Q_p}{c_p(T_f - T_0)} \Leftrightarrow T_s > T_{s,min} = T_0 + \frac{Q_p}{c_p}$$

Here we can give a physical interpretation of  $T_{s,min}$ . It is the temperature that would be achieved at the interface without any heat feedback from the gas phase. Indeed if  $d_x T(0^+) = 0$ , we can integrate Equation (3.3) from  $-\infty$  to 0 and find  $T_s = T_0 + Q_p/c_p$ . Following Assumption 10 ( $c_p = c_c$ ), we recover our previous expression of  $T_{s,min}$ .

Now we need to show that all acceptable solutions have the property  $T_s > T_{s,min}$ . The monotonicity of the temperature in the gas phase, established in Proposition 3.2.1, associated with the condition  $T_s < T_f$  shows that  $d_x T > 0$  in the gas phase. This means that  $\gamma$  is always positive in the gas phase: heat is always conducted from the gas phase into the solid phase. As a consequence,  $T_s > T_{s,min}$  is always satisfied in our problem. That is also what we would expect from a physical point of view, as we know the gas phase will actually heat up the solid, not cool it down. Moreover, using the constant combustion enthalpy property from Proposition 3.1.4 and the global energy balance from Proposition 3.1.3, we find that  $T_s > T_{s,min}$  is also the required condition to ensure  $Y(0^+) < 1$ .

Overall, we are now assured that the surface temperature  $\theta_s$  will always be higher than  $\theta_{s,min}$ . Via the pyrolysis law (3.2), the minimum surface temperature  $\theta_{s,min}$  corresponds to a regression velocity  $r_{min} > 0$ . Therefore we conclude that  $r$  will always be larger than  $r_{min}$ .

Let us now compute the value of  $\xi$  for this value of  $r$ :

$$\xi(r_{min}) = \gamma(\theta_{s,min}^+, r_{min}) - \eta\gamma(\theta_{s,min}^-, r_{min}) + \eta r_{min} \frac{Q_p}{Q_p + Q} = \gamma(\theta_{s,min}^+, r_{min})$$

We have previously shown  $\gamma(\theta, r) = r\theta$  in the solid phase, hence  $\gamma(\theta_{s,min}^-, r_{min}) = r_{min}\theta_{s,min} = r_{min}Q_p/(Q_p + Q)$ . Consequently the last two terms of the previous equation cancel each other,

leading to  $\xi(r_{min}) = \gamma(\theta_{s,min}^+, r_{min})$ . The strict monotonicity of  $\theta$  implies that  $\gamma$  is always positive. As a consequence,  $\xi(r_{min}) > 0$ . We have shown in the proof of Proposition 3.2.2 that  $\xi(r_{max}) < 0$ , therefore there exists a solution wave velocity  $r$  in the interval  $(r_{min}, r_{max})$ , such that  $\xi(r) = 0$ .

On this interval, we have established that  $d_r \xi < 0$ , whence we conclude that the solution is unique within this interval. Let us underline again that solutions outside of this interval are not physical and would lead to a violation of the monotonicity of the temperature profile. Physical interpretations of the behaviour of  $\xi$ ,  $\Delta\gamma$  and  $S$  are presented in 3.2.5.  $\square$

At this point, we have proved that there exists a unique solution to the reduced system, therefore also for the dimensionless one and for the simplified travelling wave problem which were presented in Sections 3.1.3 and 3.1.5. There exists only one steady travelling combustion wave solution for the burning of a homogeneous solid propellant with simplified kinetics and a pyrolysis concentrated at the surface, with the surface temperature being linked to the mass flow rate by a pyrolysis law such that the mass flow rate monotonously increases with the surface temperature. The proof is valid for a very wide range of values for the heat of reaction of the pyrolysis process, and for any value of the gas phase activation energy.

**Remark 8.** *Assumption H10 ( $c_c = c_p$ ) made the proof of uniqueness much easier. If we had not used it, the target interface balance  $S$  would have a more complex variation with respect to  $r$  and no easy conclusion on uniqueness would be possible. However the assumption can easily be relaxed in the numerical method presented further in this chapter, as it only changes the definition of  $S$ ,  $Q_p$  and  $T_f$  (see Section 3.3.3). It is likely that the solution may remain unique on a certain range of values for the ratio  $c_p/c_c$ , and this was indeed observed in our test case for a wide variety of values for this ratio.*

**Remark 9.** *Johnson and Nachbar [196] proved the uniqueness of  $r$  for any fixed value of  $T_s$ . This study case can also be treated with the approach we have presented, however a few adjustments are necessary, which are exposed in the Appendix of our article [66].*

**Remark 10.** *The study presented in this chapter also encompasses the laminar flame study by Zeldovich [64]. In this case,  $\theta_s = \theta_{ignition}$  is a constant, and  $S = 0$  as no chemical reactions takes place at the interface, therefore the existence and uniqueness is proved directly from Proposition 3.2.3.*

### 3.2.4 Heteroclinic orbit and critical points

The points  $x = -\infty$  and  $x = +\infty$  are critical points for the dimensionless system, i.e. all derivatives are zero. These points correspond to  $(\theta = 0, \gamma = 0)$  and  $(\theta = 1, \gamma = 0)$ . The dynamics of the dimensionless system in phase space is a heteroclinic orbit that joins these two points. This orbit and the associated treatment of the critical points is very similar to the bistable planar waves studied in [205]. The first critical point  $(0, 0)$  is more difficult to analyse, as it is not a hyperbolic point, however we can easily integrate (3.26) and find that the solution is  $\gamma = r\theta$ . The other critical point  $(1, 0)$ , in the gas phase, is a hyperbolic point, therefore the solution curve (orbit) will depart from the associated stable manifold. We can then determine the slope  $d_\theta \gamma(1)$  by means of a linearisation. We use the approximations  $\gamma(\theta) = \alpha(\theta - 1)$  and  $\Psi(\theta) = \beta(\theta - 1)$ , with  $\alpha = d_\theta \gamma(1)$  and  $\beta = d_\theta \Psi(1)$ . Following remark 4, we know that  $\beta < 0$ . Injecting these linearised expressions into (3.27), we get:  $\alpha^2 + \alpha\tilde{m} + \beta = 0$ . This second order equation has two real solutions of opposite signs. As we require

$\alpha = d_\theta \gamma(1) < 0$  so that our solution remains in the half-plane  $\gamma \geq 0$ , we find:

$$\alpha = \frac{\tilde{m}}{2} \left( 1 - \sqrt{1 - \frac{4\beta}{\tilde{m}^2}} \right) \quad (3.31)$$

The behaviour around the two critical points will be used in the numerical strategy based on a shooting method to integrate the dynamics of the orbit.

### 3.2.5 Physical interpretation and discussion

In the present chapter, we have introduced  $\Delta\gamma$ , which is the dimensionless thermal power excess that is available to power the pyrolysis process, i.e. gas heat feedback minus the thermal power used to heat up the solid, and  $S$ , the dimensionless thermal power that is required for the pyrolysis process to be sustained at the given value of  $r$ . We have shown that  $d_r \Delta\gamma < 0$ , therefore increasing  $r$  will decrease the thermal power available for the pyrolysis. The sign of  $d_r S$  indicates how the required thermal power evolves with the regression speed  $r$ .

**case  $Q_p < 0$**  In the case  $Q_p < 0$ , the pyrolysis process is endothermic, i.e. it absorbs heat from the gas and solid phases. This can be the case if the sublimation of  $G_{(s)}^1$  into  $G_{(g)}^1$  is very demanding in terms of energy, which corresponds to  $L_v > Q_s$  in Section 3.1.1. We showed that in this case,  $d_r S > 0$ . If, for an arbitrarily chosen value of  $r$ , we have  $\Delta\gamma < S$ , it means that the heat feedback from the gas phase is too low compared to the heat that would be absorbed by the solid phase and the pyrolysis reaction in a stationary state. The fact that  $d_r \Delta\gamma < 0$  and  $d_r S > 0$  shows that as we lower the mass flow rate, the thermal power excess transmitted by the gas phase to the interface increases whereas the thermal power needed for the pyrolysis decreases. As we have seen that in the limit  $r \rightarrow 0$ ,  $\Delta\gamma > S$ , we know that we will find one value of  $r$  such that both powers cancel out. Conversely if we start with  $r$  such that  $\Delta\gamma > S$ , we need to increase the value of  $r$ . The limit case, where the surface and flame temperatures are equal, i.e.  $r = (1/\rho_c)m(T_s = T_f)$  yields  $\Delta\gamma < S$ , therefore we are also ensured that we will find one solution for  $r$ .

**case  $Q_p > 0$**  The same reasoning can be applied. In this case the pyrolysis is exothermic, thus it also contributes to the heating of the solid phase. We showed that  $d_r S < 0$ , i.e. the thermal power required by the pyrolysis decreases as the mass flow rate increases, in the sense that it is actually negative and increasing in magnitude. This is physically coherent with the fact that the pyrolysis is exothermic. We have established that in the interval  $(r_{min}, r_{max})$ ,  $d_r \xi < 0$ . It shows that as we increase the mass flow rate  $m$ , the thermal power excess transmitted by the gas phase to the interface will decrease more rapidly than the thermal power needed for the pyrolysis. Therefore, starting from a value of  $r$  such that the heat feedback is too strong ( $\xi > 0$ ), lowering  $r$  will only worsen the interface thermal balance. We actually need to increase  $r$ , up until the point where the thermal power  $S$  required by the pyrolysis catches up with the thermal power excess  $\Delta\gamma$ .

The gradient jump  $[d_x \theta]_0^{0+}$  is the same as  $[\gamma]_{\theta_s(r)^-}^{\theta_s(r)^+}$ . Using (3.21), we can rewrite this as  $[\gamma]_{\theta_s(r)^-}^{\theta_s(r)^+} = S(r) + (\eta - 1)\gamma(\theta_s(r)^-)$ . In the particular case where  $\eta = 1$ , i.e. both phases have the same thermal conductivity, this reduces to  $S(r)$ , thus the gradient jump has the sign of  $S$ . If we have  $\eta \neq 1$ , the sign of the gradient jump will depend on the gradient in the solid phase at the interface. For example, if  $\eta > 1$ , the temperature gradient jump at the interface will be positive only if  $S(r) > (1 - \eta)\gamma(\theta_s(r)^-)$ .

This theoretical study brings in two aspects. First, it allows to describe a greater variety of physical scenarios, compared to the ones represented by the existing analytical models. Second, and this is the purpose of the next section, it allows for an efficient numerical resolution.

### 3.3 Numerical method and verification against a CFD code

We now explain how the previous analysis is used to construct a numerical shooting method to iteratively determine the solution, i.e. the correct wave velocity (eigenvalue) and temperature profile (eigenfunction). We also present a one-dimensional CFD code developed at ONERA for the study of solid propellant combustion. This CFD code can be adapted to use the same level of modelling as the shooting method. Our first objective is therefore to compare the results of both methods to cross-verify these tools within the framework defined by our modelling assumptions (Sections 3.3.3 and 3.3.3). Our second objective is to show how the shooting method may be extended to relax some of these assumptions and what limitations may be encountered (Sections 3.3.3 and 3.3.3). For this purpose, the CFD code will serve as a reference, as it allows for a straightforward relaxation of several assumptions.

#### 3.3.1 Shooting method

##### Determination of the phase portrait for a given $r$

For a given value of  $r$ , we can integrate the dimensionless equations (3.26) and (3.27) from the reduced system, which are first-order ODEs for the variable  $\gamma$  as a function of  $\theta$ . In the solid phase, the integration is analytical, as we directly obtain  $\gamma(\theta) = r\theta$ . This gives us the value of  $\gamma$  for  $\theta \in [0, \theta_s(r)]$ . In the gas phase, Equation (3.27) can be written as:  $d_\theta \gamma = \eta r - (\Psi/\gamma)$ . We need to integrate this equation from  $\theta = 1$  to  $\theta = \theta_s$ . As explained in subsection 3.2.4, the starting point ( $\theta = 1, \gamma = 0$ ) is a critical point for our system, therefore starting a numerical integration from this point is impossible. To overcome this problem, we simply use the linearised solution slope  $\alpha$  given in (3.31), and start the integration from  $(1 - \Delta\theta, -\alpha\Delta\theta)$ , avoiding the critical point. We typically use  $\Delta\theta = 10^{-6}$ . To maximise the accuracy, the integration of the gas phase Equation (3.27) is performed using the Radau5 algorithm [206], featuring an adaptive step size, with very tight tolerances ( $\approx 10^{-14}$ ). Once the profile of  $\gamma$  is computed, we can go back to the spatial representation by using the definition  $\gamma = d_x \theta$  to compute  $x(\theta) = \int_{\theta_s}^{\theta} \frac{z}{\gamma(z)} dz$ . This formula also ensures that  $x(\theta_s) = 0$ .

##### Determination of $r$ through a dichotomy process

Based on our analysis of  $\xi$ , we know that  $\xi(0) > 0$  and  $\xi(r_{max}) < 0$ . In the case  $Q_p < 0$ , we start a dichotomy from the two initial points 0 and  $r_{max}$ , the latter being computed beforehand from the global energy balance and the pyrolysis law. If  $Q_p > 0$ , we replace the starting value 0 with  $r_{min}$ . In both cases,  $\xi$  is monotonous between the two initial points and undergoes a change of sign, therefore convergence of the dichotomy process is ensured. For each new guess of  $r$ , we integrate the reduced system as explained previously, and obtain the value of  $\Delta\gamma(r)$ . We compare it to the value of  $S(r)$  to compute  $\xi(r)$ . Based on the sign of  $\xi(r)$ , we can shrink the interval where  $\xi$  changes sign, until the change in  $r$  between each iteration becomes small enough.

We could also perform a constrained optimisation on the variable  $r$ , minimising the objective function  $f(r) = [S(r) - \Delta\gamma(r)]^2$ , with the constraint  $r \in [r_{min}, r_{max}]$ . Practically, the optimisation method is quicker to find the approximate solution, but fails at determining  $r$  as precisely as the dichotomy process, even when using tight tolerances. However the dichotomy requires many iterations, therefore the more advanced Brent root-finding method [207] was used. In our test cases, the solution was usually found within 10 iterations.

**Remark 11.** *This semi-analytical method is bound to be more accurate than the analytical models discussed in the introduction, as these models basically use the same assumptions, but also assume that the activation energy of the gas phase reaction is either very high or zero. Our method does not need this information and will better reproduce the gas flow. This comes at the cost of having to iterate on the value of  $r$ , each time integrating numerically the reduced system. However, this cost will be rather low, as each iteration only requires the integration of the simple ODE (3.27). This method is consequently especially useful to perform extensive parametric studies.*

## Error of the method

The numerical shooting method contains 3 sources of error:

- error in the estimation of  $d_\theta\gamma(1)$ , used to avoid the critical point in the gas phase;
- error in the numerical integration of the gas phase temperature profile;
- convergence precision achieved by the shooting method on the value of  $r$ .

Let us address the different items in this list. First, a simple parametric study on the value of  $\Delta\theta$  has shown that  $d_\theta\gamma$  is a constant in the neighbourhood of the critical point. Different values of  $\Delta\theta$  have been tested and the converged regression velocities are exactly identical for all  $\Delta\theta$  lower than  $10^{-3}$ . Consequently the linearisation around the critical point is a reasonable approach and the error it produces is zero up to machine precision. The numerical integration of the gas phase with the Radau5 algorithm with very tight tolerances is also close to machine-precision, as the algorithm is of order 5 and the step size is dynamically adapted to maximise accuracy. A convergence study has been performed by varying the integration tolerance from  $10^{-3}$  to  $10^{-15}$ , each time determining the solution  $r$  via the dichotomy process (Brent method). It has been found that the solution wave velocity obtained for a tolerance of  $10^{-14}$  is converged with a relative error of  $10^{-14}$ . Finally, the dichotomy process usually is able to converge the solution  $r$  with a relative error of the order of  $10^{-15}$ . Overall the only practical limitation to the precision of this numerical shooting method is the machine accuracy chosen for implementation. All our numerical computations have been performed with double precision.

### 3.3.2 Reference CFD code

We wish to compare our semi-analytical model with a proven CFD code in a less restrictive framework. The aim is to verify the shooting method results and validate our assumptions. The CFD tool developed at ONERA is a Fortran90 code based on a finite-volume approach for the one-dimensional problem, inspired from [208]. It is a steady-state version of the code presented in Chapter 5. The model has also been adapted for the combustion of aluminium particles [209]. The molecular diffusion fluxes are approximated using a second-order central difference scheme. The convective fluxes are approximated either by a first-order upwind scheme, a second-order hybrid scheme weighted by the local Péclet number, or a second-order centred scheme. The equations are written in their steady form in the travelling combustion wave reference frame. They are discretised in space and,

together with the boundary conditions, they represent a system of coupled nonlinear equations. A modified Newton method with damping is used to determine the solution, as described in [208]. The Jacobian matrix is computed numerically by finite differences. The convergence strongly depends on the initial state. If convergence is poor, temporal evolution terms can be added to the equations to approach the steady-state solution through a number of transient iterations. This code also contains an automatic grid-refinement algorithm that ensures the mesh is fine enough in the regions where the gradient or the curvature of the solution variables are high. The refinement is performed after each successful convergence to a steady solution, until all refinement criteria are satisfied. The code can handle detailed chemistry by accessing reaction and thermodynamic data through an interface with CHEMKIN-II [210]. Detailed molecular transport with binary species diffusion is also possible with the use of the EGLIB library [168]. However for the comparison with the numerical shooting method, these additional capabilities are not used. This CFD code yields solutions which are subject to different sources of errors: the quality of the discretisation (grid refinement), the tolerance for the Newton method, and the fluxes approximations. All CFD results presented further on were computed with automatic mesh refinement criteria such that any additional refinement does not change the solution. The tolerance on the norm of the Newton step is  $10^{-8}$ , and it was verified that lowering this tolerance did not change the results.

### 3.3.3 Numerical verification and parametric studies

#### Reference case with unitary Lewis

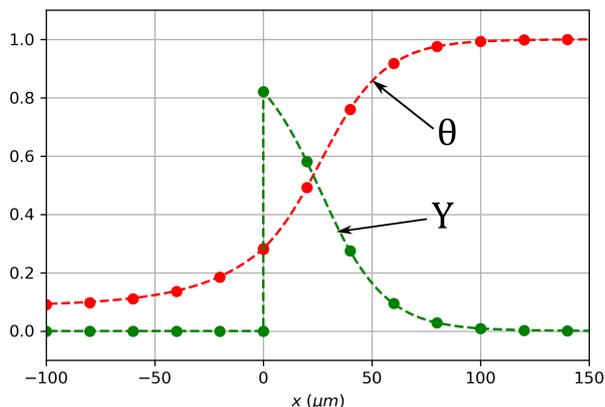
The reference case we will use throughout the rest of this chapter is the combustion of a one-dimensional equivalent of the AP-HTPB-Al propellant. The values for the different properties are adapted from [159], and are summarised hereafter.

**Solid phase** The solid phase is composed of the solid species  $P$  and has the following properties:  $\rho_c = 1806 \text{ kg.m}^{-3}$ ,  $h_p^0 = 0 \text{ J/kg}$  at  $T = 0 \text{ K}$ ,  $c_c = 1253 \text{ J/kg/K}$ ,  $\lambda_c = 0.65 \text{ W/m/K}$ . The initial temperature is  $T_0 = 300 \text{ K}$ .

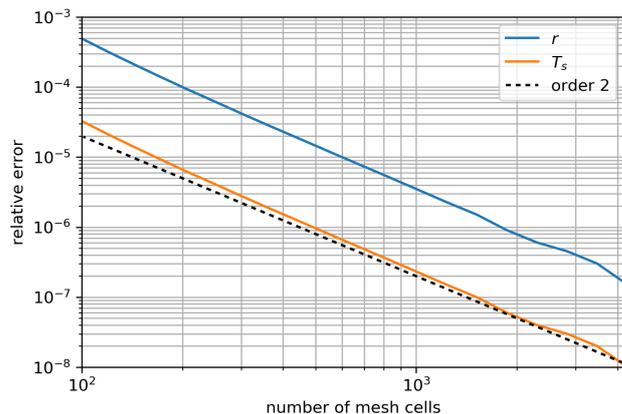
**Surface** The pyrolysis mass flow rate is computed as:  $m = A_p \exp(-T_{ap}/T_s)$ , with  $A_p = 6.07 \times 10^7 \text{ kg/s/m}^2$  and  $T_{ap} = 15082 \text{ K}$ . The pyrolysis process converts the solid phase into the gaseous species  $G_1$ .

**Gas phase** Two global species are considered: the reactant  $G_1$  and product  $G_2$ , which have the same properties except standard enthalpies. Their molar mass is  $\mathcal{M} = 74 \text{ g/mol}$ , and their heat capacities are  $c_p = c_c$ . The standard enthalpies at  $T = 0 \text{ K}$  are  $h_{G_1}^0 = -1.80 \times 10^5 \text{ J/kg}$  and  $h_{G_2}^0 = -4.06 \times 10^6 \text{ J/kg}$ . The unique global reaction is  $G_1 \rightarrow G_2$  and irreversible. The reaction rate is computed as:  $\tau = A[G_1] \exp(-T_a/T)$ , with  $A = 435.5 \text{ s}^{-1}$ ,  $T_a = 7216 \text{ K}$ , and  $[G_1] = \rho Y/\mathcal{M}$  the concentration of  $G_1$ . The diffusion coefficients are equal for both species and taken as a linear function of  $T$  such that the Lewis number is one throughout the gas phase. The thermal conductivity is  $\lambda = 0.464 \text{ W/m/K}$ .

The pressure is set to 5 MPa. The heat of reactions are  $Q_p = 1.8 \times 10^5 \text{ J.kg}^{-1}$  (independent of  $T$  since  $c_p = c_c$ ), and  $Q = 3.9 \times 10^6 \text{ J.kg}^{-1}$ . For the CFD code, the diffusion coefficient  $D_g$  of both species is taken as a linear function of  $T$ , such that the Lewis number  $\text{Le} = (\lambda_g/(\rho(T)c_p))/D_g(T)$  is 1 across the gas phase. Figure 3.3 shows a graphical comparison of the dimensionless temperature



**Figure 3.3** Plot of  $\theta$  and  $Y$  for the semi-analytical solution (dots) and the numerical simulation (dashed lines)



**Figure 3.4** Convergence of the CFD solution towards the semi-analytical solution with an adapted mesh

$\theta$  and mass fraction  $Y$  profiles. The agreement is very good, and has been verified for several other values of the pressure  $P$  (e.g. 0.5 MPa). The relative error between the regression speed obtained from the semi-analytical tool and the one obtained with the CFD code with mesh adaptation is around  $10^{-7}$ . This allows us to conclude on the verification of our numerical strategy and model implementation.

### Spatial convergence of the CFD solution

It is interesting to study the convergence in space of the steady-state CFD solution. We perform multiple simulations on increasingly refined grids. The meshes are generated as follows: knowing the temperature profile from the semi-analytical solution and starting from an initial grid point at  $x = 0$  (interface), the other grid points are placed such that the difference in interpolated temperature between two successive grid points is below a certain threshold. By varying this threshold (from 0.05K to 50K), grids with varying level of refinement are obtained, whose point distribution is relatively well adapted to the problem. The finite volume mesh is then generated by taking these grid points as the positions of the cell faces. In this reference case, the thermal layer in the solid phase and the flame in the gas phase both have a thickness close to  $10^{-4}$ m. The generated meshes are extended by adding cells with gradually increasing sizes so that the abscissa of the outer cells are ten times greater than this thickness in order to minimise the influence of the Neumann boundary conditions. It has been verified that extending the mesh further does not improve the relative error.

We show in Figure 3.4 the convergence of the CFD result towards the semi-analytical solution for the reference case. We see that second-order accuracy is reached, and that the relative errors reach  $10^{-8}$  on  $T_s$  ( $10^{-7}$  on  $r$  and similar results are obtained on temperature profiles) at around 4000 adapted mesh cells. For a given level of relative error, it was determined that a uniform mesh would require approximately ten times more points when using the smallest cell size of the corresponding adapted mesh. This shows that the CFD code is definitely more computationally intensive, and requires an adapted mesh to produce accurate results. Achieving a relative error lower than  $10^{-8}$  on  $T_s$  is difficult as this level of error is very close to the tolerance on the Newton step, i.e. the relative precision of the CFD solution obtained by the Newton solver.

Overall, the error is sufficiently small so that we can consider that the CFD solution is converged in terms of spatial mesh and Newton iterations. The automatic mesh refinement available in the CFD tool yields similar level of errors, therefore it will be used for the rest of the comparison. The resolution of the travelling wave problem is coherent between the two tools thus bringing out a useful cross-verification of both approaches.

### Parametric study with variable gas phase activation energy

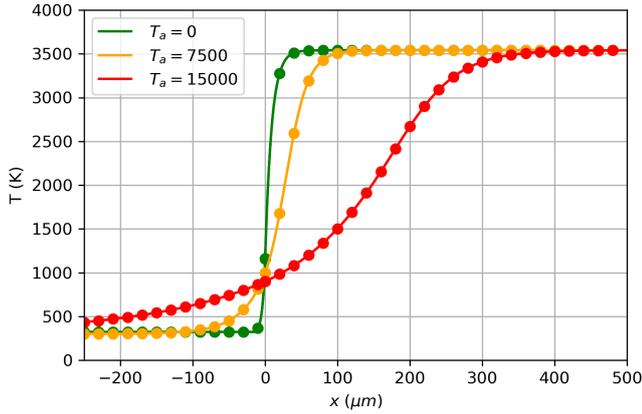
We know that for the simplified chemical mechanism used, the activation energy  $E_a$  of the gas phase reaction will be of paramount importance. Indeed, if  $E_a$  is low, the reaction will be very fast at lower temperatures in a narrow zone just above the surface, which will lead to a strong heat feedback and a high regression rate. On the opposite, if it is very large and every other parameter is unchanged, the reaction will be slower and more spread out spatially, thus diminishing the heat feedback from the gas phase onto the solid propellant, resulting in a slower regression rate.

To highlight the effect of  $E_a$ , we compute with both methods the temperature profile for three values of  $T_a = E_a/R$  (activation temperature), representative of low, mid and high activation energies. The pressure remains at 5 MPa. The Lewis number is 1 for both methods. All the other parameters are not modified, therefore neither the regression velocity, the surface temperature, nor the heat feedback from the flame will be the same for all three cases.

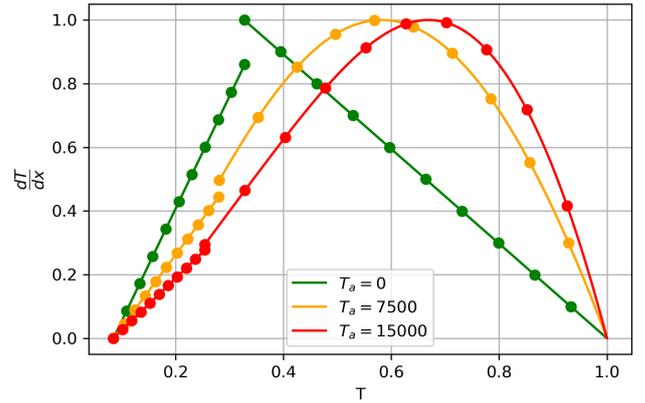
Figure 3.5 shows the spatial temperature profiles. We see that as  $T_a$  decreases, the profile becomes sharper and the flame gets closer to the surface of the propellant. Figure 3.6 shows the phase portraits of these three simulations. The ordinate  $d_x T$  is scaled for each simulation separately, so that the maximum is 1, otherwise the high values of  $d_x T$  encountered in the case  $T_a = 0$  would make it difficult to compare the curves. As  $T_a$  increases, the abscissa  $T_s$ , i.e. the propellant surface temperature, increases and so does the height of the gradient jump between the two phases. As the activation energy is lowered, the flame becomes thinner and the heat feedback on the solid grows due to the stronger temperature gradient near the surface. The higher surface temperature results in a greater regression rate through the pyrolysis law (3.2), which in turn increases the thermal effect of the pyrolysis, i.e. the size of the gradient jump at the surface.

The fact that the gas phase portrait for  $T_a = 0$  is a straight line can be surprising. This is actually related to the Arrhenius law used. As stated before, the reaction rate is of the form  $\tau \propto [G^1]T \exp(-T_a/T)$ . Using the constant enthalpy from Proposition 3.1.4, the ideal gas law and expressing the concentration  $[G^1]$  as  $\rho Y/\mathcal{M}$  in Equation (3.6), one may easily verify that a linear function of the form  $\gamma = \alpha(1 - \theta)$  is a solution.

A more thorough parametric study has been performed to obtain Figure 3.7. The agreement of both methods for the prediction of the regression speed is very good across the whole range of activation temperatures, with a relative error of approximately  $10^{-7}$  on  $r$ . An important remark is that the CFD solution often fails to converge when the initial mesh is not suited, and when the initial solution is not sufficiently good. For example, the case  $T_a = 0$  involves very strong temperature gradients, which required adding many more mesh points close the surface for the initial solution. On the opposite, the case  $T_a = 15000$  K gives a very smooth and slowly evolving temperature profile, but this translates to a very spread out flame, requiring additional mesh points far from the surface so that the combustion process is fully represented. Rather than remedying these problems manually by using a single extended mesh and performing transient iterations to facilitate convergence, we use the semi-analytical method to generate the initial solution, and define an initial mesh as explained in Section 3.3.3 with a sufficient extension so that the gas phase reaction is completed within the computational domain. With this approach, the CFD code converges very



**Figure 3.5** Temperature profiles, CFD results (full lines) compared to semi-analytical results (circles)



**Figure 3.6** Normalised phase portraits, CFD results (full lines) compared to semi-analytical results (circles)

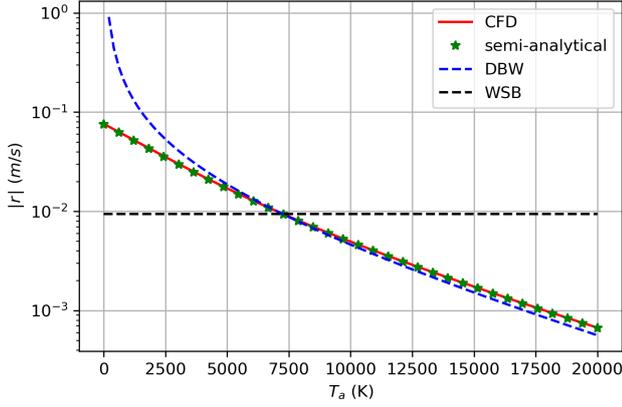
quickly and can further refine the mesh if needed. This highlights some of the main advantages of the semi-analytical method, which are that the solution always converges, and that no manual mesh adjustments are needed.

Figure 3.7 also shows the results of the analytical models WSB and DBW. Their pre-exponential factors  $A_p$  and  $A$  were adjusted so that both models predict the same regression velocity at  $T_a = 7216$  K as the CFD model. The WSB model assumes  $T_a = 0$ , therefore the regression velocity does not vary with  $T_a$ . We see that the tendencies are reasonable, even if not in perfect agreement (we use a log scale), between the semi-analytical model and the DBW model for high activation energies. However the DBW model, which assumes high gas phase activation energy, falls apart when  $T_a$  is decreased. Overall, the semi-analytical model is a more generic model that produces quantitatively good results, without any assumption on  $T_a$ .

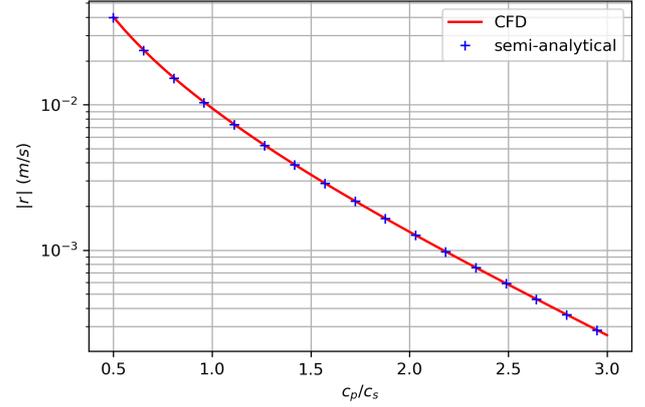
### Parametric study on the ratio $c_p/c_c$

We now wish to extend the numerical method beyond its theoretical ground, by relaxing Assumption H10. Simulations are performed with the CFD code and our semi-analytical tool, by varying the gas specific heat  $c_p$  at constant  $c_c = 1.2 \times 10^3 \text{ J.kg}^{-1}\text{K}^{-1}$ . The species diffusion coefficient  $D_g$  is taken as a linear function of  $T$ , such that the Lewis number remains equal to unity. Therefore it varies with  $c_p$ . In our semi-analytical tool, we account for  $c_p \neq c_c$  by using Equation (3.14) instead of (3.25) for  $S$ , and  $T_f$  and  $Q_p$  are computed as in remarks 2 and 10. The ODE (3.27) changes slightly as the ratio  $c_c/c_p$  appears as an additional factor for the term in  $\eta r$ , which also affects the slope (3.31) of the solution near the critical point. The results are shown in Figure 3.8 for a wide range of ratios  $c_p/c_c$  (0.5 to 3), which encompasses all physically relevant solid propellant configurations. We see that the CFD code and the semi-analytical model are again in very close agreement. The relative error between both tools is around  $10^{-8}$  on the surface temperature ( $10^{-7}$  on  $r$ ).

As we do not have a theoretical proof of existence and uniqueness when  $c_p \neq c_c$ , we have performed a more extensive study to observe the behaviour of  $\xi$  when we vary the ratio  $c_p/c_c$ , even if it is outside of the physically relevant interval. The curve of  $\xi(r)$  is plotted for various ratios  $c_p/c_c$  in Figure 3.9a. Each curve is normalized by  $\xi(0)$ , the limit of  $\xi$  when  $r$  tends to 0. We see that  $\xi$

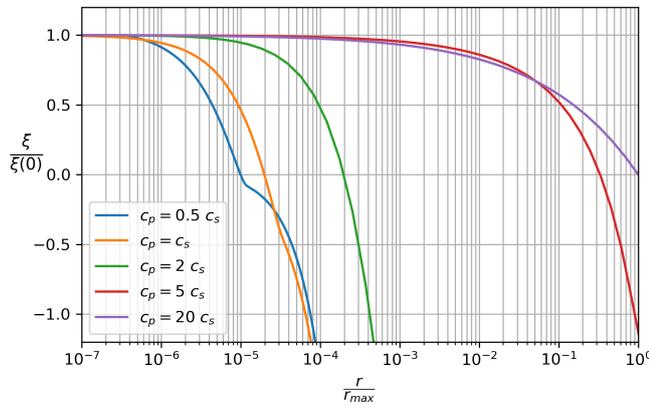


**Figure 3.7** Regression speed as a function of activation temperature, CFD results compared to semi-analytical and analytical results

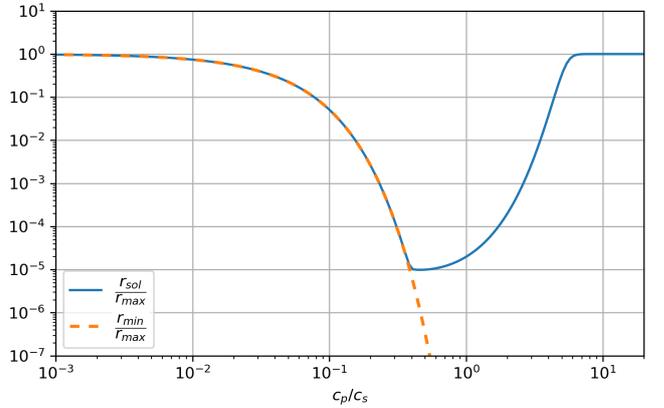


**Figure 3.8** Evolution of the regression speed with  $c_p/c_c$  obtained with the semi-analytical and CFD methods

remains monotonous and only has one zero-crossing. Figure 3.9b shows how the solution wave speed  $r_{sol}$  is located between the bounds  $r_{max}$  and  $r_{min}$  as the ratio  $c_p/c_c$  changes. We observe that the solution remains within these bounds, and tends to  $r_{max}$  for high values of the ratio  $c_p/c_c$ . When this ratio is low, both the solution and  $r_{min}$  tend to  $r_{max}$ . Overall, this numerical investigation shows that the semi-analytical model can still be reliable beyond the simplified level of modelling adopted for the theoretical analysis.



(a) Evolution of  $\xi$  with  $r$  for different ratios of  $c_p/c_c$

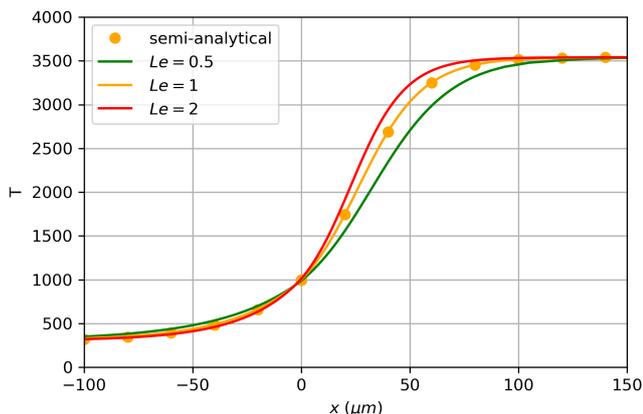


(b) Evolution of the solution regression speed  $r_{sol}$  (blue dashes) and lower bound  $r_{min}$  (orange line) compared to  $r_{max}$

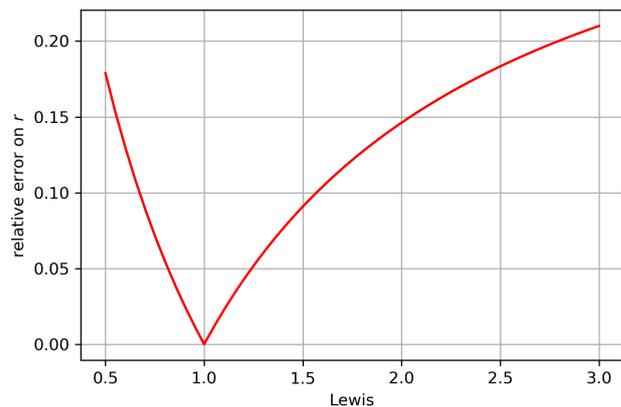
**Figure 3.9** Effects of the ratio  $c_p/c_c$  on  $\xi$  and on the solution regression speed

### Parametric study on the Lewis number

The unitary Lewis number assumption H8 allows to simplify the system by only having to consider the temperature and its gradient as variables. This can be relaxed numerically, however it would



**Figure 3.10** Temperature profiles, CFD results (full lines) compared to semi-analytical results (dots) for different values of the Lewis number



**Figure 3.11** Relative error on the prediction of  $r$  by the semi-analytical method compared to the CFD method, as a function of the Lewis number

make the shooting method more complex, requiring the addition of the mass fraction  $Y$  and its gradient as variables in the dynamical system equations. It may also affect the existence and uniqueness of the solution. Zeldovich has reported that uniqueness can indeed be lost for laminar flames [64] when  $Le > 1$ . To show the limits of the semi-analytical model, we conduct a study on the effect of a constant Lewis number, but with a value different than 1. To do so, just as before, the CFD code uses species diffusion coefficients that are linear with  $T$ , such that  $Le$  is constant in the gas phase.

If  $Le$  is high, heat diffuses faster than species, therefore we expect a stronger thermal feedback from the gas phase, resulting in a faster regression speed. When the Lewis number is decreased below 1, we expect the opposite effect. This is confirmed in Figure 3.10 which shows the temperature profiles for three different values of  $Le$ . Figure 3.11 shows the relative error of the semi-analytical model for the estimation of  $r$ , compared to the CFD result, for Lewis numbers within the realistic range from 0.5 to 3. As expected, the minimum error is reached at  $Le = 1$ . For  $Le > 1$ , the semi-analytical model underestimates  $r$  as it underestimates the temperature gradients near the surface. For  $Le < 1$ ,  $r$  is overestimated. Still, the relative error on  $r$  lies within 20%. The relative error on  $T_s$  is much smaller ( $\approx 1\%$ ). The exponential term in the pyrolysis law with the high pyrolysis activation temperature  $T_{ap}$  is the root of this difference, as a small relative error in the evaluation of  $T_s$  translates into a greater one for  $r$ . Overall, this parametric study shows that the unitary Lewis number assumption still allows for a quantitatively reasonable solution.

### 3.4 Nonlinear behaviour of $Q_p$

To further improve the ability of the semi-analytical model to reproduce more complex behaviours, a nonlinear dependence of the surface heat release  $Q_p$  on  $T_s$  can be considered. In this section, we restrain ourselves to the case  $c_p = c_c$  for simplicity. Let us assume the following form of  $Q_p$ :

$$Q_p(T_s) = Q_p^{adiab} + g(T_s)$$

with  $Q_p^{adiab}$  the heat of the pyrolysis reaction in the case without nonlinearity (constant here as  $c_p = c_c$ ). The global energy balance then reads:

$$T_{f,nonlinear} = T_f^{adiab} + \frac{g(T_s)}{c_p} \quad (3.32)$$

with  $T_f^{adiab} = T_0 + (Q_p^{adiab} + Q)/c_p$  the flame temperature obtained without the nonlinearity.

This allows the inclusion of various effects, such as:

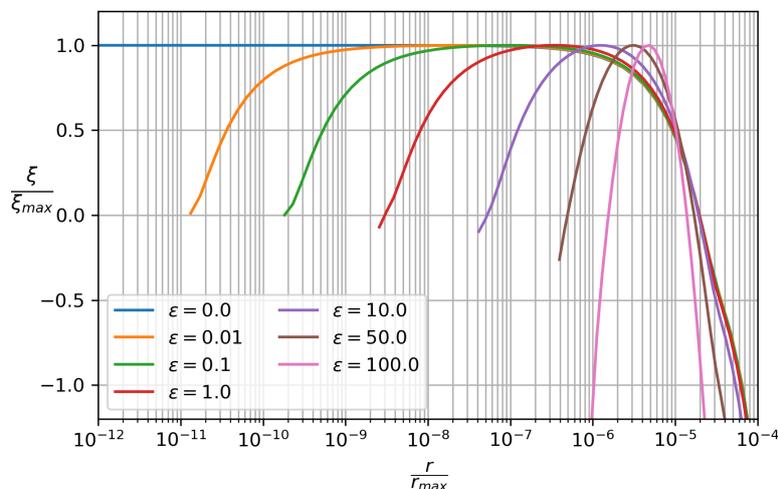
- a constant laser heat flux  $q_{laser}$  with  $g(T_s) = \frac{q_{laser}}{m}$
- heat loss by radiative emission from the surface (suggested in [200]) with  $g(T_s) = \frac{-\epsilon\sigma T_s^4}{m}$ , where  $\sigma = 5.670374 \times 10^{-8} \text{W}\cdot\text{m}^{-2}\text{K}^{-4}$  is the Stefan-Boltzmann constant, and  $\epsilon$  the surface emissivity.
- a linear or nonlinear dependence of  $Q_p$ , achieved by any nonlinear form for  $g$ , without dividing by  $m$ .

In terms of non-dimensional quantities, the effect of  $g$  is held within the function  $S(r)$ .

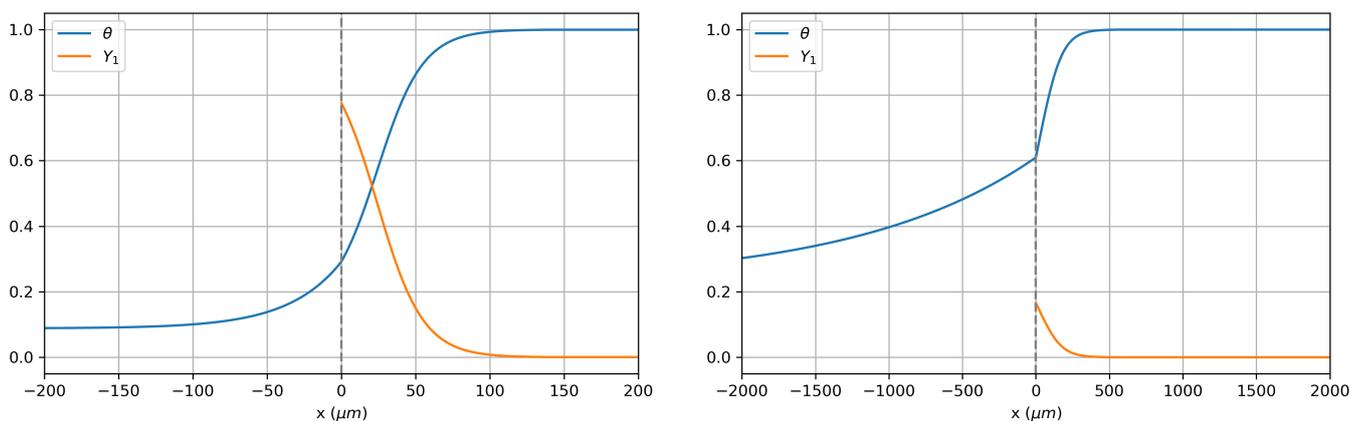
Johnson and Nachbar [200] have shown that the addition of a radiative heat loss at the surface allows for the appearance of a second solution to the travelling wave problem. We have been able to numerically reproduce this finding. We show in Figure 3.12 the evolution of  $\xi$  with  $r$  for various values of  $\epsilon$ . Each curve is only drawn on the range of acceptable regression velocities  $r$  such that  $T_s \leq T_f$ . Each curve of  $\xi$  is normalised by its maximum value, and the abscissa are normalised by the maximum regression speed  $r_{max}$ , which depends on  $\epsilon$ . Values of  $\epsilon$  higher than 1 are non-physical, however they allow for a better visualization of both solutions. Here, we do not intend to perform a physical interpretation of the results, but only aim at studying how the system behaves with such nonlinearities.

We see that, when  $\epsilon$  is high enough, the curve  $\xi$  intersects the horizontal line  $\xi = 0$  twice. Each of these intersections corresponds to a solution wave speed  $r$  with an associated solution profile. The rightmost intersection is very close to the original solution obtained without radiative losses (adiabatic case, curve with  $\epsilon = 0$ ), and the solution is plotted in Figure 3.13a. Near these relatively high regression speeds, the mass flow rate  $m$  varies very rapidly with  $T_s$ , as we can see if we use the usual exponential form  $m(T_s) = A_p \exp(-T_{ap}/T_s)$ . If we increase  $T_s$ , the increase in  $m$  is consequently much more important than the increase of the radiative heat loss proportional to  $T_s^4$ . Therefore, when  $\epsilon$  takes reasonable values, this leads to a relative absence of effect of the heat loss by radiation in this region of high mass flow rates, and the behaviour of  $\xi$  is locally the same as in the adiabatic case. This changes however in the low regression speed region where  $T_s$  is low, as the mass flow rate  $m$  becomes much lower than 1, which emphasises the heat loss effect in  $g$  to such an extent that we may find a second steady state solution (see Figure 3.13b). Note that in this case, the heat loss is very important in comparison with the heats of reaction, and this translates into a lower final temperature  $T_f$  as compared to the adiabatic case. Physically, we know that in the adiabatic case,  $\xi$  is positive when  $r$  is small, that is the heat feedback from the gas phase is too strong compared to the thermal power actually needed to maintain the solid phase temperature profile and the pyrolysis process. It is then clear that adding a heat loss at the surface may be able to evacuate the excess heat feedback.

We have observed that the second solution disappears or at least is out of the bounds on  $r$  (such that  $T_s < T_f$ ) when  $\epsilon$  is too low. On the opposite, as  $\epsilon$  increases, we see in Figure 3.12 that both solutions come closer to each other, up to a point where they merge (for  $\epsilon \approx 162.8$  in our case). If we increase  $\epsilon$  further, no solution exists any more, and  $\xi$  remains negative, i.e. the gas phase is not able to transfer enough heat to the surface and solid phase. Visually, although not shown in the previous Figure, the parabolic curve of  $\xi$  obtained for  $\epsilon = 100$  is also observed for higher values of  $\epsilon$ , but its highest point gradually descends towards negative values, crossing zero for  $\epsilon \approx 162.8$  as both solutions merge before disappearing. These curves were not plotted as no coherent normalization could be found to allow for an easy visual comparison of the different cases. With the chosen model parameters, the secondary solutions obtained for physically acceptable values of  $\epsilon$  (lower than 1) correspond to very low regression rates and cannot be considered as physically realistic.



**Figure 3.12** Evolution of  $\xi$  in the case of heat loss by radiation at the surface, for different values of  $\epsilon$ .



(a)  $T_s = 989$  K,  $m = 14.6$  kg/s,  $T_f = 3392$  K

(b)  $T_s = 803$  K,  $m = 0.42$  kg/s,  $T_f = 1317$  K

**Figure 3.13** Solutions obtained with  $\epsilon = 50$

It would be interesting to study the stability of the second solution (where radiative effects

dominate), as Johnson and Nachbar [200] expect it should lead to extinction. Note that finding these multiple solutions is relatively straightforward with the semi-analytical tool, whereas it would not be as simple with a CFD code.

Overall, more complex behaviours can be captured within the semi-analytical model. Many aspects however become more difficult with the addition of nonlinearities (bounds on  $r$ , behaviour of  $\xi$ , existence and uniqueness). Still, surface nonlinearities with sufficiently low contributions may be easily added without impacting the convergence of the algorithm.

## Partial conclusion

We have presented a new method for the determination of travelling wave solutions for a simplified combustion model of a homogeneous solid propellant. The main assumptions are that the gas phase only contains one reactant and one product, the reactant being transformed into the product by a single irreversible reaction, and that the Lewis number is 1. Considering solutions of this nonlinear eigenvalue problem in the form of a travelling wave profile as well as a regression velocity  $r$ , we have derived a reduced system which can be used in a numerical shooting method to determine the correct regression velocity. We have proven that the travelling wave solution profile and velocity exist and are unique under conditions which are not restrictive in view of the physical properties encountered in real solid propellants.

A numerical comparison has been conducted with a CFD code developed at ONERA, and the agreement is very good for a broad range of parameter values, at least for a unitary Lewis. The shooting method is simpler to implement, more efficient and reliable than the CFD code. We have shown that the relative error on  $r$  grows as the Lewis number changes, but the semi-analytical solution remains quantitatively correct for realistic values of this parameter. A comparison has also been made with some of the main analytical models, and we have shown that our semi-analytical model produces better results overall. Practically, the semi-analytical method is free of any space discretisation error. The remaining sources of error are well controlled. This method can thus be a useful verification tool for CFD codes with simplified test cases. Besides, the method always converges, hence it can be used to generate initial solutions for more detailed methods that would otherwise struggle to converge.

This method can also be employed to determine the various coefficients needed to compute the linear response function to pressure fluctuations, by performing multiple simulations with slight variations of one parameter.

The proof of existence and uniqueness may be extended to include the effect of a constant external heat flux absorbed both at the surface and inside the solid. Care must be taken, as the burnt gas temperature  $T_f$  will depend on the mass flow rate. Also, as in [196], a reverse reaction may be included in the gas phase, allowing for a non-trivial equilibrium far away from the surface.

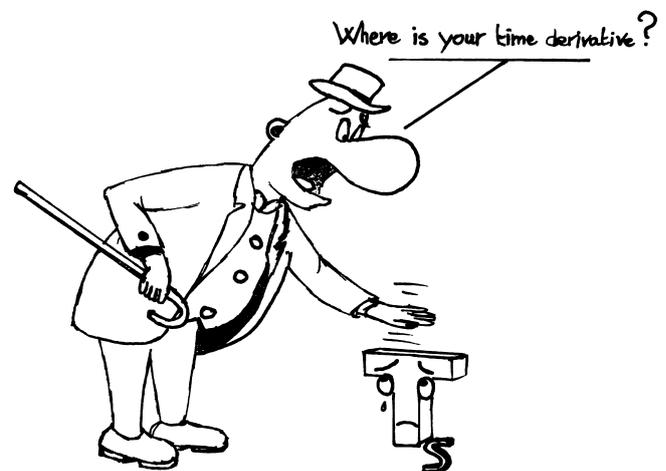
We have conducted a numerical experiment on the inclusion of nonlinear surface phenomena. For example, following the work of Johnson and Nachbar [200], a radiative heat loss on the propellant surface can be included. This heat loss is accounted for in  $S$  and also brings a new dependence of  $T_f$  on  $r$ . It is observed that  $\xi$  is not a monotonous function any more: depending on the surface emissivity, there can be one solution, two, or none. When there are two solutions, only one is thought to be stable, but the shooting method is still able to find the unstable one. This shows the potential of the semi-analytical tool. When the shooting method is extended to account for such phenomena, the existence and uniqueness of the solution might not be guaranteed any more, and the bounds on the regression velocity may need to be redefined. Finally, the dichotomy process

may need to be performed on multiple separate intervals for the regression velocity to allow the determination of all solutions.

A few evolutions can be envisioned for the numerical shooting method. More advanced pyrolysis laws can be used, as discussed in Section 2.3. For instance, if the physical configuration is such that the law from Equation (2.51) results in a mass flow rate that increases with  $T_s$ , then the theory from this chapter is directly applicable. Otherwise, some adjustments may be necessary. Radiation penetration and absorption, as well as heat loss by thermal radiation and potential heat loss to the surroundings may be easily integrated into the numerical tool. This would require a numerical resolution of the solid phase temperature profile, as already done for the gas phase, instead of the simple analytical solution that we have been able to use in this chapter. The shooting method may also be adapted to take into account a temperature dependent Lewis number, using the same approach as Zeldovich in [64] for laminar flames. This requires the addition of  $Y$  and  $\delta = d_{\bar{x}}Y$  as variables in the gas phase, and also a reassessment of solution's slope near the critical point at  $\theta = 1$ . Note however that, already in the simpler case of homogeneous laminar flames, Zeldovich has shown that uniqueness of the solution may be lost [64].

# Chapter 4

## Semi-discretisation in space and mathematical nature of the obtained system



### Summary

A semi-discretisation of the unsteady model from Chapter 2 is performed via the method of lines and a finite volume approach. The resulting system of equations is shown to be of differential-algebraic nature, with an index equal to 1. The knowledge of that nature will be of great importance for the choice of the numerical strategy in the Chapter 5.

The previous mathematical analysis in steady-state has shed light on the importance of the surface coupling conditions. These conditions already require an iterative process for a solution to be obtained in the previous steady-state setting. It is therefore sensible to expect that these conditions will also require a careful handling in the unsteady case.

In the present chapter, we come back to the generic unsteady model, with the aim of developing a numerical tool to perform simulations of one-dimensional combustion transients such as laser-induced ignition. Semi-discretisation in space of the generic model is performed via the use of a finite volume approach. The obtained system of semi-discrete equations must then be integrated in time, as typically done with a set of ordinary differential equations. However, we show that this system is actually of differential-algebraic nature, causing many issues with typical time schemes used in the combustion community. A careful survey of the schemes available in the literature

leads to the choice of a family of high-order adaptive implicit Runge-Kutta methods, particularly well-suited for our system.

This mathematical analysis and the associated choice of time scheme is an important contribution to the one-dimensional low-Mach homogeneous combustion and one-dimensional solid propellant combustion communities, clearly indicating the mathematical and numerical requirements for high-fidelity simulations of such problems.

The content of this chapter and the two subsequent ones is adapted from one of our articles [69].

## 4.1 State of the art for the one-dimensional low-Mach combustion

Solving for unsteady homogeneous and spray or solid-propellant combustion in one-dimensional flame simulations has attracted enormous attention in the combustion community starting with the seminal work conducted by M.D. Smooke and collaborators between Sandia and Yale University [208, 211–214]. To our knowledge, in these studies and subsequent papers, most of the one-dimensional unsteady CFD codes for such applications use a time integration based on splitting and/or implicit methods, which are usually limited to first-order accuracy in time. No mathematical analysis has been reported regarding the nature of the system of coupled equations obtained after semi-discretisation in space, where the handling of some variables (e.g. surface temperature for the solid propellant case, eigenvalues such as strain rate or mass fluxes) requires careful examination in connection with boundary conditions. Besides, relying on low-order integration methods may prove disadvantageous in terms of accuracy, performance and ability to resolve fine dynamics. Indeed high-order methods are especially important when investigating instabilities and nonlinear behaviours, e.g. limit cycles, where growth of some modes can only be captured by high-fidelity numerical methods [41]. In the homogeneous combustion literature, one notable exception is the work on the dynamics of non-premixed counterflow flames by Im et al. [40]. In this interesting piece of work, high-order time integrators for differential-algebraic equations (DAE) were introduced, but the constraint formulation was rather involved (introduction of compressibility effects - link with the index) and the details of the convergence / efficiency were not the main purpose of the paper.

Even if solid propellant combustion brings in additional difficulties and constraints related to the heterogeneous nature of the flow, homogeneous combustion or two-phase flow combustion for one-dimensional low-Mach flows involve in fact the same problematic, and the results provided in this work can equally be applied to these other fields of research. It is therefore instructive to conduct a short overview of the various numerical approaches presented in the literature for the time integration of one-dimensional solid propellant combustion models, which is typical of what we have described in the general field of 1D low-Mach 1D combustion. One of the earliest detailed one-dimensional model is presented by Erikson and Beckstead [36]. A splitting method is implemented to integrate the gas phase equation, using the ICE scheme [215] to compute the pressure and velocity fields with an implicit scheme of first-order accuracy in time. The stiff chemical source terms are handled with DVODE [216]. The solid phase energy equation is integrated implicitly. The surface temperature and regression speed are then iterated upon until the interface conditions are met, each time performing the split integration of both phases. Due to poor computational performance and large splitting errors, they transition in [145] to a fully implicit resolution of the gas phase, using the TWOPNT [217] algorithm to solve the system discretised in time with the first-order implicit Euler method. The authors mention the attempt to use DASSL [218] instead, which is a high-order

adaptive method based on a backward-differentiation formula, however difficulties led them to use TWOPNT most of the time. Both phases still need to be iterated upon at each time step. V. Yang and his colleagues (see the review [18]) used a similar approach with an iterative coupling of both phases. The solid phase species equations, the gas phase continuity and species equations are solved explicitly with the fourth-order RK4 method, while the remaining equations are solved with PREMIX [211], which uses the first-order implicit Euler scheme. A variant of this approach [146] solves the gas phase implicitly with a dual time step to improve the convergence of the Newton solver, however implicit Euler is still the method used for the time integration. Meredith and Beckstead [34] and Smith [33] simulate the one-dimensional ignition of various propellants using the same iterative approach and the same resolution method for the condensed phase as Erikson and Beckstead, but with the ability for the solid phase to take multiple smaller steps to improve convergence. The gas phase is solved implicitly with a Newton algorithm and time integration methods provided in the PETSC library [219]. No precision is given on the order and properties of the chosen methods. To our knowledge, most of the one-dimensional unsteady CFD codes use a similar iterative procedure between the different phases, with a time integration that is overall first-order accurate in time, as discussed at the beginning of the introduction.

In the scope of this work, there is a convergent need for a one-dimensional solid propellant combustion solver, either for the coupling with a 3D CFD tool as discussed in Chapter 7 for the simulation of SRM ignition, or for the parametric / detailed study of flame dynamics and unsteady combustion dynamics in a purely one-dimensional context. This solver, based on the generic unsteady model from Chapter 2 should have a tailored numerical strategy, enabling high-order in time as well as adaptive time stepping, to ensure that the simulation results are accurate and efficiently obtained. As we will see in this chapter, specific care is required to take into account the fact that the model involves algebraic constraints.

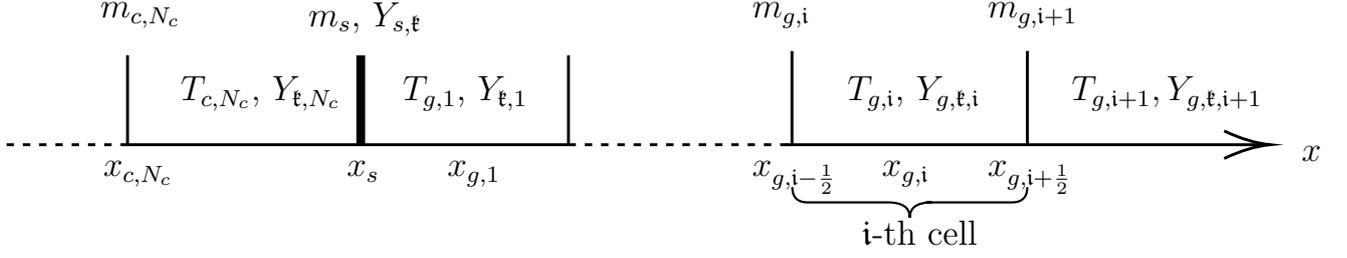
In the present chapter, we first semi-discretise in space the generic unsteady model, using classical finite volume techniques. We then investigate our generic unsteady model from Chapter 2 and show that semi-discretisation in space leads to a system of index-1 DAEs, on which some theoretical background is given. The knowledge of this particular property will be decisive when looking for high-order time integration methods. Many suited methods are reported in the literature for this particular class of problems [206, 218]. It is important to stress the fact that, when the problem is considered in more than one dimension, the system may become differential-algebraic of index two, in particular due to the pressure field [218], requiring a deeper analysis of the problem and more advanced time integration methods [220]. However, the main results of the one-dimensional case should remain applicable.

## 4.2 Semi-discretisation in space of the unsteady model

For the numerical implementation of our solid propellant combustion model, we apply the method of lines to obtain a set of discrete evolution equations. The semi-discretisation in space is obtained with a finite volume approach, however other approaches could be applied without affecting the conclusions from the present chapter.

### 4.2.1 Gas phase

The set of conservative equations for the gas phase is semi-discretised in space with a finite volume approach: the domain is split in  $N_g$  cells (control volumes). The discretised variables are the



**Figure 4.1** Localisation of the discretised variables in the finite volume mesh. The thick vertical line represents the surface.

temperature  $T$ , the mass fractions  $Y_{\xi}$ , the mass flow rate  $m = \rho u$ . The temperature and mass fractions are taken at the centers of each cell, while the mass flow rate is taken at the left face of each cell. This staggered-grid approach helps decreasing the numerical discretisation error, and is convenient for the free boundary problem with a flux defined at the surface. The surface temperature  $T_s$ , the surface mass fraction of the  $\xi$ -th species  $Y_{s,\xi}$  are taken at  $x = 0$  (rightmost face of the solid domain, leftmost face of the gas domain). The localisation of each variable is sketched in Figure 4.1.

Using the notation  $q$  to identify any of the conservative variables:  $\rho Y_{\xi}$  and  $\rho h$ , or  $\rho$ , and the subscript  $i$  as the index of the mesh cell considered, the conservative equations (2.9) to (2.11) become:

$$\Delta x_i \frac{dq_i}{dt} = - [F_{d,q} + F_{c,q}]_{i-\frac{1}{2}}^{i+\frac{1}{2}} + \Delta x_i s_{q,i} \quad (4.1)$$

with  $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$  the size of the corresponding cell,  $F_{d,q}$  the diffusive fluxes,  $F_{c,q}$  the convective fluxes and  $s_{q,i}$  the source term.

Thermodynamic and transport properties are evaluated at the cell centers, and their values at the interfaces are taken as averages of the adjacent cells values. The gradient at the  $(i - \frac{1}{2})$ -th interface of a variable  $q$  discretised at the cell centers is computed as:

$$\nabla q_{i-\frac{1}{2}} = \frac{q_i - q_{i-1}}{x_i - x_{i-1}} \quad (4.2)$$

The interface values of the transported variables are computed via a numerical scheme. For any conservative variable  $q$ , its interface value at the abscissa  $x_{i-\frac{1}{2}}$  is defined as:

$$q_{i-\frac{1}{2}} = \Phi_{i-\frac{1}{2}}^- q_{i-1} + \Phi_{i-\frac{1}{2}}^+ q_i \quad (4.3)$$

where  $\Phi_{i-\frac{1}{2}}^-$  and  $\Phi_{i-\frac{1}{2}}^+$  are the scheme coefficients and sum up to one. If both these coefficients are equal to 0.5, the scheme is centered. This scheme is the only second-order accurate scheme with a 2-point stencil. MUSCL-type schemes need at least a 3-point stencil. Even though the centered scheme is theoretically stable for convection problems when used in conjunction with an implicit time integration, issues can still arise. In particular, there can be a loss of diagonal dominance for the Jacobian used in the Newton algorithm to compute the solution values at each step [221],[222, page 446]. This leads to error accumulation in the solution of the linear system (e.g with a Thomas algorithm). Also, nonlinear instabilities may arise [223]. In our experience, we have indeed seen that our solution method struggles to converge when the centered scheme is used on large cells where convection dominates. To circumvent these issues, the centered scheme needs to be locally upwinded if the flow is convection-dominated, ensuring convergence at the expense of falling back to

first-order accuracy. This is done dynamically via a Péclet-weighted average of the first order upwind and second order centered schemes, similarly to what is done in [222, page 448] with the concept “mesh Reynolds number”. The local Péclet number at the center of the  $i$ -th cell is  $Pe_i = c_{p,i} m_i / \lambda_i$ , where the reference length is chosen unitary. The Péclet number at the  $(i - \frac{1}{2})$ -th interface is computed as:

$$Pe_{i-\frac{1}{2}} = \frac{1}{2} (Pe_i + Pe_{i-1}) (x_i - x_{i-1}) \quad (4.4)$$

i.e. it is an average Péclet number with a reference length taken as the distance between the centers of the neighbouring cells. The Péclet number is a measure of the relative importance of diffusive phenomena compared to convective transport, here for the enthalpy. If  $|Pe_{i-\frac{1}{2}}| < 0.5$ , we use the centered scheme:  $\Phi^+ = \Phi^- = 0.5$ , i.e. the centered scheme is used as the thermal diffusion is locally the dominating form of energy transfer compared to convection. In that case, the strong diffusion dampens the instability caused by the lack of upwinding of the convective terms. If  $|Pe_{i-\frac{1}{2}}| > 1$ , we use the upwind scheme:  $\Phi^+ = 0$  if  $Pe > 0$ , else  $\Phi^+ = 1$ . The transition between these two cases is smooth with respect to  $Pe_{i-\frac{1}{2}}$ , so as not to cause numerical issues later on. This allows to use a second-order accurate scheme where the mesh is sufficiently refined. The first-order scheme is only used in poorly resolved areas far away from the surface, where precise representation of the flow is not required. Convective fluxes are then computed as:

$$F_{c,q,i-\frac{1}{2}} = m_i q_{i-\frac{1}{2}} \quad (4.5)$$

Diffusive fluxes are approximated by a second-order centered scheme. Source terms are evaluated at the cell centers. The final semi-discrete equations for the evolution of the conservative variables in the  $i$ -th cell read as follows:

$$\Delta x_i \frac{d\rho_i}{dt} = m_i - m_{i-1}, \quad \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \quad (4.6)$$

$$\begin{aligned} \Delta x_i \frac{d\rho_i Y_{\ell,i}}{dt} = & + m_i \left( \Phi_{i-\frac{1}{2}}^+ Y_{\ell,i} + \Phi_{i-\frac{1}{2}}^- Y_{\ell,i-1} \right) - m_{i+1} \left( \Phi_{i+\frac{1}{2}}^+ Y_{\ell,i+1} + \Phi_{i+\frac{1}{2}}^- Y_{\ell,i} \right) \\ & + J_{\ell,i-\frac{1}{2}} - J_{\ell,i+\frac{1}{2}} + \Delta x_i \omega_{\ell,i} \quad \forall \ell \in \llbracket 1, n_e \rrbracket \end{aligned} \quad (4.7)$$

$$\begin{aligned} \Delta x_i \frac{d\rho_i h_i}{dt} = & \Delta x_i d_t P + m_i \left( \Phi_{i-\frac{1}{2}}^+ h_i + \Phi_{i-\frac{1}{2}}^- h_{i-1} \right) - m_{i+1} \left( \Phi_{i+\frac{1}{2}}^+ h_{i+1} + \Phi_{i+\frac{1}{2}}^- h_i \right) \\ & - \frac{\lambda_{i-1} + \lambda_i}{2} \frac{T_i - T_{i-1}}{x_i - x_{i-1}} + \frac{\lambda_i + \lambda_{i+1}}{2} \frac{T_{i+1} - T_i}{x_{i+1} - x_i} \\ & + \sum_{\ell=1}^{n_e} \left( \frac{h_{\ell,i} + h_{\ell,i-1}}{2} J_{\ell,i-\frac{1}{2}} - \frac{h_{\ell,i+1} + h_{\ell,i}}{2} J_{\ell,i+\frac{1}{2}} \right) \end{aligned} \quad (4.8)$$

## 4.2.2 Solid Phase

The solid phase energy equation (2.7) is replaced by a conservative equation for the enthalpy and discretised in the same way as the conservative equations in the gas phase. The solid mesh contains  $N_c$  cells. As explained further in Section 5.1.4, the block-tridiagonal Jacobian of the complete system leads to an improved computational efficiency. In order to keep a consistent Jacobian structure between the gas and solid phases, the mass flow rate field  $m$  and the species profile  $Y_{\ell}$  are kept as dummy variables. As the solid propellant is assumed inert and incompressible, the continuity equation is equivalent to  $d_x m = 0$  which is discretised as  $m_i = m_{i+1}$ , with the boundary

condition  $m_{N_c+1} = m(T_s, P)$ . Species evolution is not considered in the solid phase, therefore the simple equation  $Y_{\mathfrak{i},\mathfrak{i}} = 0$  is used, for all  $\mathfrak{i} \in \llbracket 1, N_c \rrbracket$ , and  $\mathfrak{k} \in \llbracket 1, n_e \rrbracket$ .

### 4.2.3 Surface

The surface variables are the surface temperature  $T_s$  and the surface mass fractions  $Y_{s,\mathfrak{k}}$  on the gas side. The surface matching conditions (2.21), (2.24) and (2.27) and the continuity equation (2.9) are discretised as follows, with a first-order approximation of the gradients:

$$\begin{cases} 0 = g_{th} := -\lambda_c \frac{T_s - T_{c,-1}}{x_s - x_{c,1}} + \lambda \frac{T_{g,1} - T_s}{x_{g,1} - x_s} + m(T_s) Q_p(T_s) & (4.9) \\ 0 = g_{sp,\mathfrak{k}} := m(T_s)(Y_{inj,\mathfrak{k}} - Y_{s,\mathfrak{k}}) + J_{s,\mathfrak{k}} \quad \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket & (4.10) \end{cases}$$

with  $T_{c,-1}$  the temperature in the last cell of the solid phase below the surface, and  $T_{g,1}$  the temperature in the first cell of the gas phase, just above the surface. The species surface diffusion fluxes are computed as:

$$J_{s,\mathfrak{k}} = \rho(T_s, Y_s, P) D_{\mathfrak{k}}^{eq}(T_s, Y_s, P) \frac{Y_{g,\mathfrak{k},1} - Y_{s,\mathfrak{k}}}{x_{g,1} - x_s} \quad (4.11)$$

where  $D_{\mathfrak{k}}^{eq}$  is the equivalent diffusion coefficient computed from Equation (2.13).

## 4.3 Differential-algebraic nature of the semi-discretised system

We aim at developing a 1D unsteady CFD tool for high-fidelity simulations of transient phenomena. Relying on a finite-volume space discretisation, we have obtained a system of equations on discrete variables. In this section, we show that a difficulty arises from the nature of this system: some variables are not defined by differential, but only through algebraic equations. The system thus belongs to the class of Differential-Algebraic Equations (DAE). We refer the reader to [206, 218] for details on DAEs and only the necessary aspects of this class of problem will be recalled here.

### 4.3.1 Identification of the constraints

The discretised surface variables  $Y_{s,\mathfrak{k}}$  and  $T_s$  only appear in Equations (4.9) and (4.10), however no time derivative appear. Such variables are called algebraic and will “instantly” adapt to the variations of the other variables in the cells adjacent to the surface, i.e. they are not directly affected by their time histories.

The remaining algebraic equations come from the discrete gas continuity equation (4.6), which we recall here:

$$\begin{cases} 0 = g_{m,1} := m_1 - m(T_s) & (4.12) \\ 0 = g_{m,i} := \frac{d\rho_i}{dt} + \frac{m_i - m_{i-1}}{x_i - x_{i-1}} \quad \forall i \in \llbracket 2, N_g \rrbracket & (4.13) \end{cases}$$

Equation (4.12) is the boundary condition for the gaseous mass flow rate field and is equivalent to equation (2.21).

The density  $\rho_i$  in the  $i$ -th cell is a function of  $T_i$  and  $Y_{\mathfrak{i},i}$  via the ideal gas law (2.19). These variables are governed by the discretised form of Equations (2.10) and (2.26) and by the pressure

$P$ , assumed spatially constant and given as an input to the model. Therefore  $\rho_i$  is not a true differential variable. Its time derivative  $d_t \rho_i$  appearing in equation (4.13) is entirely determined from the variations of the other gas-phase variables  $T_i$  and  $Y_{\xi,i}$ . The continuity equation is solely used to constrain the flow rate field in the gas phase  $m$ , the time derivative of which does not appear in this one-dimensional low-Mach model. As a consequence, the discrete values of the mass flow rate are also algebraic variables, which adapt instantly to variations of the other variables in coherence with the parabolic nature of the low-Mach number limit, where all pressure waves propagate at infinite speed and are relaxed instantly. This situation is generic in low-Mach number combustion modelling.

### 4.3.2 Analogy with a singular perturbation problem

To better understand the origin of the appearance of this algebraic character, and for the sake of the example, let us focus on the surface temperature. If we relax the tacit assumption that no accumulation of mass or energy takes place at the surface, we can derive a differential equation for  $T_s$ . Let us for instance assume that the interface is a thin volume of thickness  $\delta x$  with uniform temperature  $T_s$ , heat capacity  $c_s$  and density  $\rho_s$ , and is able to accumulate energy. The interface energy condition (2.27) becomes:

$$d_t(\rho_s c_s T_s) = \frac{1}{\delta x} [-\lambda_c \partial_x T(0^-) + \lambda \partial_x T(0^+) - m(T_s) Q_p(T_s)] \quad (4.14)$$

If we consider the surface properties  $\rho_s$  and  $c_s$  constant, and introduce the parameter  $\epsilon = \rho_s c_s \delta x$ , we obtain:

$$\epsilon d_t T_s = -\lambda_c \partial_x T(0^-) + \lambda \partial_x T(0^+) - m(T_s) Q_p(T_s) \quad (4.15)$$

When  $\delta x$  tends to 0 (infinitely thin interface), we perform a singular perturbation on the previous equations ( $\epsilon \rightarrow 0$ ). The time derivative of  $T_s$  disappears and the interface thermal balance reduces to equation (2.27). The equation, originally of the differential type, becomes algebraic; this is a classical result of singular perturbation theory, which is also valid for the surface mass fractions  $Y_{s,\xi}$ . This simplified analysis helps to clarify the origin of the algebraic character of Equations (2.24) and (2.27): no accumulation of mass or energy in the infinitely thin interface.

The algebraic nature of the gas-phase mass flow rate  $\rho u = m$  comes from the low-Mach limit assumption. Indeed the thermodynamic pressure is uniform in space, and its value is prescribed to the model, either as user-specified constant, or following an externally dictated evolution, for example given by a 0D chamber model as in Section 1.2.3. From the ideal gas law, we see that the knowledge of  $P$  and of the temperature profile suffices to determine the density field. Therefore, the continuity equation simply becomes a constraint on the velocity field and the momentum equation is not required any more in this one-dimensional context, unless the field of pressure perturbation is sought.

Note that in more than one dimension, the continuity equation remains scalar and involves the various components of the velocity field. Hence it is not sufficient to determine the latter. The momentum equation needs to be reintroduced in the model, giving back a differential nature to the density and velocity fields. The pressure perturbation then needs to be accounted for. It acts as a Lagrange multiplier, i.e. forcing field, to ensure the continuity equation is satisfied. In that case, the only algebraic field is this pressure perturbation, however the index (defined in the next section) of the overall system is increased to 2, causing additional difficulties for the numerical resolution.

### 4.3.3 Some background on DAEs

Before going further in the analysis of the algebraic character of the model from Chapter 2, let us make a brief overview of differential-algebraic equations and their specificities. The general form of an autonomous (i.e. without explicit time dependence) DAE is:

$$F(y, y') = 0 \quad (4.16)$$

with  $y \in \mathbb{R}^n$  and  $F : (\mathbb{R}^n, \mathbb{R}^n) \rightarrow \mathbb{R}^n$ , assuming proper initial conditions. The non-autonomous case can be recovered simply by adding the time  $t$  as an additional component of  $y$  whose time derivative is set to 1. ODEs are special cases of DAEs. For instance, the simple ODE  $x' = \lambda x$  can be written in the DAE form  $x' - \lambda x = 0$ .

In the particular case where the Jacobian  $\partial_{y'} F$  is constant, we can build a function  $G$  such that we can reformulate Equation (4.16) as:

$$(\partial_{y'} F) y' = G(y) \quad (4.17)$$

and the term  $\partial_{y'} F$  is commonly referred to as the mass matrix  $M$  of the system. If this matrix is nonsingular, the previous equation correspond to a system of ODEs. If this matrix is singular, at least some of the variables become algebraic.

A particular form of DAEs is obtained with  $M = \text{diag}(1, \dots, 1, 0, \dots, 0)$ . The original vector of variables  $y$  can then be split in two vectors. The first one corresponds to the non-zero diagonal elements of  $M$ , which we refer to as  $y$  as well for simplicity. This vector is governed by an ODE. The second vector corresponds to the zero diagonal elements of  $M$ , and we denote it as  $z$ . In that case, Equation (4.16) can be reformulated as:

$$y' = f(y, z) \quad (4.18)$$

$$0 = g(y, z) \quad (4.19)$$

with  $y \in \mathbb{R}^{n_d}$ ,  $z \in \mathbb{R}^{n_a}$ ,  $f : (\mathbb{R}^{n_d}, \mathbb{R}^{n_a}) \rightarrow \mathbb{R}^{n_d}$  and the constraint function  $g : (\mathbb{R}^{n_d}, \mathbb{R}^{n_a}) \rightarrow \mathbb{R}^{n_a}$ . It is referred to as *semi-explicit* or *Hessenberg* because the time derivatives of  $y$  are explicitly stated, while those of  $z$  do not appear. Therefore, the variables that are governed by algebraic equations are clearly identified. Such a DAE form often arises in engineering problems. As in any other form of DAEs, the difficulty lies in solving the previous system for the values of  $z$ . In particular, Equation (4.19) does not give any information on the temporal variation of  $z$ . To let the time derivative  $z'$  appear, we can derive the Equation (4.19) with respect to time:

$$0 = (\partial_y g) y' + (\partial_z g) z' \quad (4.20)$$

If the Jacobian  $(\partial_y g)$  is invertible, we can directly determine  $z'$  as:

$$z' = -(\partial_z g)^{-1} (\partial_y g) f(y, z) \quad (4.21)$$

Note however that this is not a perfect solution in practice, because deriving the constraint with respect to time may be cumbersome, and computing  $z$  by numerically integrating  $z'$  is prone to error accumulation, i.e. the value of  $z$  will eventually drift away from the set  $\{z \in \mathbb{R}^{n_a}; g(y, z) = 0\}$ . A corrective procedure would be to project  $z$  regularly onto the previous set [206].

In the case where  $(\partial_y g)$  is not invertible, one may derive  $p$  times Equation (4.19) with respect to time, until the system:

$$y' = f(y, z), \quad 0 = g(y, z), \quad 0 = \frac{dg}{dt}(y, z), \quad \dots, \quad 0 = \frac{d^p g}{dt^p}(y, z) \quad (4.22)$$

allows to construct an explicit formulation for  $g'$ , i.e. the DAE system can be converted to an ODE system, referred to as the *underlying ODE*. The integer  $p$  is then referred to as the *differentiation index* of the DAE. It is used to classify the different types of algebraic equations and is essential in order to choose the proper numerical method [206, 218]. Note that some components of  $z$  may be of lower index than others, i.e. their time derivatives may be expressed by performing less than  $p$  temporal derivations of the constraints.

In the previous case where  $(\partial_y g)$  is invertible, the index is 1, because a single derivation with respect to time of the original constraint yields an ODE on  $z$ . Index-1 DAEs are commonly encountered in mechanical systems for instance, and their resolution, although slightly more involved than that of ODEs, remains relatively straightforward, only requiring a few careful choices, as discussed later in Chapter 5. Higher-index DAEs are a much more difficult matter and often require a manual reformulation of the problem via differentiation with respect to time of certain equations to construct an index-1 equivalent problem.

#### 4.3.4 Index of the semi-discrete system

Having presented the basic concept of the index of a DAE, we now come back to our semi-discretised problem with the aim of computing the corresponding index. Algebraic equations obtained via singular perturbations are generally of index 1, therefore following the reasoning of Section 4.3.2 we expect  $T_s$  and  $Y_{s,\xi}$  to be defined by index-1 algebraic equations.

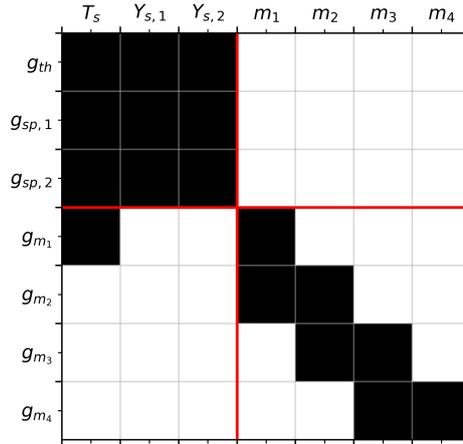
We can verify that it is indeed the case by deriving our discretised equations (4.9), (4.10), (4.12) and (4.13) with respect to time. We see that terms in  $d_t T_s$ ,  $d_t Y_{s,\xi}$  and  $d_t m_i$  appear, and that an explicit expression can be obtained for each of them. For instance,  $d_t T_s$  is obtained by deriving Equation (4.9):

$$d_t T_s = \frac{\lambda_c \frac{d_t T_{c,-1}}{x_s - x_{c,1}} + \lambda \frac{d_t T_{g,1}}{x_{g,1} - x_s}}{\frac{\lambda_c}{x_s - x_{c,1}} + \frac{\lambda}{x_{g,1} - x_s} - (\partial_{T_s} m Q_p)} \quad (4.23)$$

It is possible that the denominators present in those ODEs, e.g. Equation (4.23), become singular for certain values of  $T_s$ ,  $Y_{s,\xi}$  and carefully chosen forms of terms like  $Q_p$ ,  $m...$ . Then, the system may be locally of index 2 or higher. Such a singularity was never found in practice, therefore we do not consider this possibility further and assume that the ODEs obtained by derivation of the surface connection conditions (4.9) and (4.10) are well-posed and valid. Therefore, the associated algebraic surface variables  $T_s$  and  $Y_{s,\xi}$  are algebraic of index 1.

Regarding the gas phase continuity equation, if we differentiate Equation (4.13) with respect to time, the second-derivative  $d_{tt} \rho_i$  appears. It can however be expressed by differentiating the ideal gas law and the other conservation equations, such that a similar reasoning can be applied, leading to the conclusion that the discrete mass flow rate field  $m_i, i \in \llbracket 2, N_g \rrbracket$  is a vector of index-1 variables.

Another approach is to reorganise our system by clearly separating the algebraic equations from the differential ones. Let  $W$  be the vector of differential variables, containing the discretised values of  $T$  and  $Y_\xi$  in each cell, and let  $Z$  be the vector containing our algebraic variables  $T_s$ ,  $Y_{s,\xi}$  and



**Figure 4.2** Sparsity pattern of the Jacobian of  $g$ , the vector of constraints

$m_i$ :  $W = (T, Y_{\dagger})^t$  and  $Z = (T_s, Y_{s,\dagger}, m_i)^t$ . The system of DAEs can be recast into the following semi-explicit form:

$$\begin{cases} d_t W = f(W, Z) & (4.24) \\ 0 = g(W, Z) & (4.25) \end{cases}$$

with  $g = (g_{th}, g_{sp,1} \dots g_{sp,n_e}, g_{m_0} \dots g_{m_{N_g}})^t$  the vector of algebraic equations. In this form, we have seen previously that the index of the DAE is 1 if the Jacobian  $\partial_Z g$  is non-singular. We can verify this by forming this Jacobian, however for the sake of simplicity, we only show in Figure 4.2 its sparsity pattern when considering only 2 species. The labels on the vertical axis describe the constraint being derived, and the labels on the horizontal axis denote the differentiation variable. We can decompose the matrix into smaller blocks following the red lines. The first block on the diagonal corresponds to the Jacobian of the nonlinear system  $(g_{th}, g_{sp,1} \dots g_{sp,n_e})^t = 0$ . A numerical investigation has shown that in all studied cases, the corresponding solution is unique and the Jacobian, invertible. The second block on the diagonal is lower triangular with non-zero elements on the diagonal, therefore it is invertible. Overall we can claim that the Jacobian can be considered as invertible, hence the index of the corresponding algebraic equations is 1.

As previously stated, for readability reasons, we have used the thermal interface balance (2.27) involving the temperature, instead of Equation (2.23) involving the enthalpy. This however does not change the nature of the system.

We can draw a parallel with the eigenvalue found when investigating the steady-state problem from Chapter 3. In the steady-state problem, the unitary Lewis assumption allows us to drop the mass fractions in the gas phase and at the interface, as they are directly related to the temperature through enthalpy conservation. The continuity equation simply tells us that the mass flow rate in the gas phase is uniform and equal to the pyrolysis mass flow rate given by the pyrolysis law. The thermal interface condition (2.27) in the steady case determines the single value of the regression speed  $r$  at the interface, or equivalently the surface temperature  $T_s$ , since both are linked through the pyrolysis law (2.25), and we thus call it the eigenvalue of the problem. In the unsteady case, the constraints involve a larger number of surface variables (the surface mass fractions  $Y_{s,\dagger}$  are added to the surface temperature), but also lead to a single regression speed, which can still be

called an eigenvalue of the problem. The semi-discretised mass conservation equation (4.6) leads to a vector of algebraic variables, the discretised mass flow rate values in all the domain, which are directly determined from the regression rate and constitute a vector of eigenvalues evolving with time and uniquely determined as a function of time. This parallel allows shedding some light on the unsteady problem as another nonlinear eigenvalue problem. This analysis can be extended in a straightforward manner to the case of homogeneous or spray flames, even counterflow diffusion flames with algebraic variables involving mass constraint and strain rate.

Now that we know our semi-discrete system has an index equal to 1, we proceed in the next chapter with the presentation of solution methods and the choice of well-suited time integration schemes.



# Chapter 5

## Numerical strategy

### Summary

Following the fact that the semi-discretisation of the unsteady model leads to a system of index-1 DAEs, well-suited solution methods are presented, and a class of stiffly accurate Runge-Kutta methods is selected for implementation in the 1D simulation tool VULC1D. Particular care is devoted to the handling of the continuity equation to improve the efficiency of the numerical resolution.

### 5.1 Requirements for the time integration method

We have shown that our semi-discretised problem is an index-1 differential-algebraic system of equations. With the aim of ensuring a highly accurate resolution of the dynamics of the unsteady model, we are looking for a high-order time integration method. This will also enable a higher computational performance, as high-order methods are known to perform better than low-order ones when accurate solutions are sought [224]. We also wish to maximise the performance of the final simulation tool, thus we also search for a method that enables the use of adaptive time stepping. In this chapter, we focus on Runge-Kutta methods and shortly discuss the applicability of other methods.

#### 5.1.1 Runge-Kutta methods applied to ODEs

Let us first present a brief overview of Runge-Kutta methods and their applications to simple ODE problems. A number of properties useful in that framework will be equally important in the case of DAEs.

#### Principle

Let us consider the following scalar ODE:

$$dy = f(y), \quad y(0) = y_0 \quad (5.1)$$

with  $f$  assumed as smooth as necessary. The temporal integration of such a system typically consists in the computation of successive steps from  $t_n$  to  $t_{n+1} = t_n + \Delta t$ . Starting from the value  $y(t_n) = y_n$ ,

the exact solution  $y(t_{n+1})$  is:

$$y(t_{n+1}) = y_n + \int_{t_n}^{t_{n+1}} f(y(a)) da \quad (5.2)$$

Following the assumed regularity of  $f$ , we can rewrite the previous equation using a Taylor expansion of  $f$  with an infinite sum. If we truncate the series after  $p$  terms, we get:

$$y(t_{n+1}) = y_n + \sum_{k=0}^{p-1} \frac{\Delta t^{k+1}}{(k+1)!} \frac{d^k f}{dy^k}(y_n) + \mathcal{O}(\Delta t^{p+1}) \quad (5.3)$$

This truncated expression is an approximate solution  $y_{n+1}$ , with an error, also referred to as the local truncation error, which is  $\mathcal{O}(\Delta t^{p+1})$  in the case of a vanishing  $\Delta t$ . Therefore the approximation is said to be of order  $p$ . In practice, constructing such a numerical approximation requires the evaluation of the successive derivatives of  $f$ , which may be cumbersome or simply intractable. Therefore, other approaches have been developed which only require evaluating  $f$  at different times. Note that the above solution obtained with a truncated Taylor series is an approximation of the integral in Equation (5.2). Consequently, any other method of approximating the integral is applicable, such as quadrature formulae. In particular, the family of Runge-Kutta (RK) methods is based on the introduction of  $s$  intermediate time points between  $t_n$  and  $t_{n+1}$ :

$$t_{ni} = t_n + c_i \Delta t, \quad c_i \in [0, 1], \quad i = 1..s \quad (5.4)$$

The solution  $y_{ni}$  at each intermediate point (also called *stage*) is then constructed as:

$$y_{ni} = y_n + \Delta t \sum_{j=1}^s a_{ij} f(y_{nj}) \quad (5.5)$$

Then, the solution  $y_{n+1}$  is obtained by performing a quadrature on the previous stages.

$$y_{n+1} = y_n + \Delta t \sum_{j=1}^s b_j f(y_{nj}) \quad (5.6)$$

The precision of the quadrature (5.6) can be obtained by comparing the Taylor expansion of the quadrature formula with that of the exact solution. The method is of order  $p$  if the first term in  $y_{n+1} - y(t_{n+1})$  is of the form  $\mathcal{O}(\Delta t^{p+1})$ . The Taylor expansion enables order conditions to be derived, which must be satisfied by the method's coefficients  $a_{ij}$ ,  $b_i$ ,  $c_i$ . The method is said to be consistent if its order  $p$  is higher than 0. All these considerations only apply to the local truncation error, i.e. the error generated by a single time step. A RK scheme is said to be stable if a small perturbation during one step is not amplified in the following steps. A method that is both stable and consistent can be shown to be convergent, i.e. the global error  $y_n - y(t_n)$  at time  $t_n = T$  obtained with the initial condition  $y(0) = y_0$  after  $n = T/\Delta t$  steps, decreases with  $\Delta t$ . Thus, a refinement of the time step leads to a more accurate solution. More details on these aspects can be found in the reference book [224].

A RK method is said to be explicit if the summation in Equation (5.5) only goes up to  $k = i - 1$ , i.e. if each stage can be determined from the values of the previous stages. The matrix  $A = (a_{ij})$  is then strictly lower triangular. Otherwise, the method is referred to as implicit, and a nonlinear system composed of Equation (5.5) for each stage must be solved for  $(y_{n1}, \dots, y_{ns})$ .

## Stability

Applying a RK scheme to an ODE does not ensure that a correct result is obtained. In particular, it is well known that the numerical solution obtained with some RK methods may become unstable if the time step is too large. This phenomenon can be studied by considering the following generic vector ODE:  $d_t y = f(t, y)$ ,  $y \in \mathbb{R}^n$ . Let us assume that it possesses a smooth solution  $\phi(t)$ . If we linearise the ODE around the neighbourhood of  $\phi$ , we obtain:

$$d_t y = f(t, \phi(t)) + \left( \frac{\partial f}{\partial y} \right) (t, \phi(t)) [y - \phi(t)] + \dots \quad (5.7)$$

Introducing  $\bar{y}(t) = y(t) - \phi(t)$  and keeping the first-order terms, we can reformulate this into:

$$d_t \bar{y} = (d_y f) (t, \phi(t)) \bar{y} = J(t) \bar{y} \quad (5.8)$$

Let us assume that the Jacobian  $J(t)$  is constant. We arrive at:

$$d_t \bar{y} = J \bar{y} \quad (5.9)$$

Let us now assume that  $J$  can be diagonalised, with its eigenvalues being  $\lambda_1, \dots, \lambda_n$ . Thus  $J = P^{-1} \Lambda P$ , with  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  and  $P$  an invertible matrix. We can then rewrite our previous ODE as:

$$d_t \bar{y} = P^{-1} \Lambda P \bar{y} \quad \Leftrightarrow \quad d_t P \bar{y} = \Lambda P \bar{y} \quad (5.10)$$

Performing a variable change from  $\tilde{y} = P \bar{y}$  and dropping the  $\tilde{\cdot}$  notation for simplicity, we obtain:

$$d_t y = \Lambda y \quad (5.11)$$

In that form, each component of  $y$  evolves independently of the others, i.e. they form a series of simple scalar ODEs  $d_t y_i = \lambda_i y_i$ ,  $i \in [1, n]$ . Consequently, it is very convenient to study the numerical integration of the complete system by considering the simpler scalar ODE:

$$d_t y = \lambda y \quad (5.12)$$

with  $\lambda \in \mathbb{C}$  the eigenvalue of the scalar system. If the numerical integration is stable for all the eigenvalues of  $J$ , then the integration of the linear vector ODE (5.9) is stable, and experience has often shown that the original nonlinear ODE can also be integrated in a stable manner. Equation (5.12) is known as Dahlquist's test equation [225].

Let us apply an  $s$ -stage RK scheme to this ODE, starting with a known the initial condition  $y_n$ . Equations (5.5) and (5.6) can be reformulated in matrix form:

$$Y = y_n \mathbf{e} + \lambda \Delta t A Y \quad (5.13)$$

$$y_{n+1} = y_n + \Delta t b^T Y \quad (5.14)$$

with the Butcher matrix  $A = (a_{ij}) \in \mathbb{M}_s(\mathbb{R})$ , the vectors  $b = (b_1, \dots, b_s)^T \in \mathbb{R}^s$ ,  $\mathbf{e} = (1, \dots, 1)^T \in \mathbb{R}^s$  and  $Y = (y_{n1}, \dots, y_{ns}) \in \mathbb{R}^s$ . Under the assumption that  $\lambda \Delta t \rho(|A|) < 1$  with  $\rho(|A|) = \max(\text{eig}(|A|))$  the spectral radius of  $|A|$ , it can be shown that Equation (5.13) has a unique solution:

$$Y = (\mathbb{I} - \lambda \Delta t A)^{-1} y_n \mathbf{e} \quad (5.15)$$

Reinjecting this expression in Equation (5.14), and introducing  $z = \lambda\Delta t$ , we obtain:

$$y_{n+1} = R(z)y_n, \quad \text{with } R : z \rightarrow 1 + zb^t (\mathbb{I} - zA)^{-1} \mathbf{e} \quad (5.16)$$

Comparing this with the exact solution of Dahlquist's equation,  $y(t_{n+1}) = y_n \exp(z)$ , we see that  $R$  must approximate the exponential. Then, the order of the approximation is the order of the method.

Note that the exact solution is stable as long as  $Re(\lambda) \leq 0$ . This may however not be the case for the numerical solution obtained with a RK scheme. Indeed, the numerical integration can only be stable if  $|R(z)| \leq 1$ . Let us for instance consider the explicit Euler method:  $y_{n+1} = y_n + \Delta t f(y_n)$ . We directly get  $R(z) = 1 + z$ . Stability is ensured if  $|R(z)| \leq 1$ , i.e. if  $z$  lies within the disc of radius 1, centered on  $z = -1$  in the complex plane. Therefore, only a small portion of the “physically” stable configurations will indeed be stable when numerically integrated with this method. A desirable property, introduced by Dahlquist [225], is that the method does not suffer from such unphysical instabilities:

**Definition 5.1.1.** A method is called *A-stable* if its stability domain  $\{z \in \mathbb{C}; |R(z)| \leq 1\}$  contains the whole left half plane  $\mathbb{C}^- = \{z; Re(z) \leq 0 \text{ or } z = \infty\}$ .

For a RK method, it can be shown that  $R$  may be expressed as:

$$R(z) = \frac{\det(\mathbb{I} - zA + z\mathbf{e}b^t)}{\det(\mathbb{I} - zA)} \quad (5.17)$$

We see that the denominator is 1 for an explicit method, because  $A$  is lower-triangular. Thus the stability function is a polynomial function of  $z$ , hence the stability domain is bounded: no explicit method can be A-stable. For an implicit method, the denominator in Equation (5.17) may be of higher-order than the numerator, hence allowing for the possibility of an unbounded stability domain which may include  $\mathbb{C}^-$ . For instance, the backward Euler method  $y_{n+1} = y_n + \Delta t f(y_{n+1})$  corresponds to  $R(z) = \frac{1}{1-z}$ . The corresponding stability domain is therefore much larger than for the explicit Euler method.

A-stability is however not always sufficient. In the case of a system of equations, the eigenvalues  $\lambda_i$  may be spread over a large domain. In particular, it is very common to have eigenvalues with a very large negative real part, whereas the other eigenvalues are much closer to 0. The corresponding solution modes are very fast and stable, i.e. they very quickly converge to 0. Quite often, these modes are not of interest for the end result, and only the solution variation associated with the slower modes is sought. When that is the case, the ODE is termed as *stiff*. With explicit Euler, the time step to ensure a stable solution should typically be  $\Delta t \leq 1/\max(|Re(\lambda_i)|)$ , which can be extremely low compared to the time scales of the other modes which are of interest, resulting in a stability-constrained and very inefficient numerical integration. Applying an A-stable implicit method can improve on this, because fast modes are associated with  $|R(\lambda_i \Delta t)| < 1$ . However, in the limit of very rapid modes, i.e.  $Re(\lambda) \Delta t \ll -1$ , some A-stable methods have  $|R(z)|$  very close to 1, i.e. the fast modes are very slowly damped from one step to the next, whereas they would instantaneously vanish in the true solution. For instance, the Crank-Nicolson method [226],  $y_{n+1} = (1/2)(f(y_n) + f(y_{n+1}))$ , yields  $R(z) = \frac{2+z}{2-z}$ . Its stability domain is exactly  $\mathbb{C}^-$ , hence the method is A-stable. However  $\lim_{|z| \rightarrow \infty} R(z) = -1$ , therefore very rapid modes are not damped, and the associated solution components actually oscillate instead of converging to 0.

Very stiff ODEs are ODEs which exhibit very fast solution modes that do not need to be finely resolved to obtain an accurate reproduction of the dynamics of the slower ones. These ODEs are

therefore poorly integrated if we only require A-stability for the integration method. From the previous considerations, we see that methods for which the fast modes are associated with a value  $|R(z)| \ll 1$  would be beneficial, as these modes would be damped rapidly by the numerical method, closely reproducing the dynamics of such modes in the true solution. Therefore, the following property was introduced [227]:

**Definition 5.1.2.** A method is *L-stable* if it is A-stable and its stability function satisfies:

$$\lim_{|z| \rightarrow \infty} R(z) = 0$$

L-stability can be obtained from A-stability if the coefficients of the Runge-Kutta method obey the following definition:

**Definition 5.1.3.** An  $s$ -stage method is called *stiffly accurate* if the last stage is identical to the quadrature stage, i.e. if:

$$a_{sj} = b_j, \quad j = 1..s$$

If  $A$  is invertible, we get from (5.16) that  $R(\infty) = 1 - b^t A^{-1}$ . The stiff-accuracy property leads to  $A^t(0, \dots, 0, 1) = b$ , thus  $R(\infty) = 1 - (0, \dots, 0, 1)\mathbf{e} = 0$ . Therefore an A-stable stiffly-accurate method with an invertible matrix  $A$  is L-stable. The Crank-Nicolson method is stiffly accurate, however its Butcher matrix  $A$  is not invertible, and the method is clearly not L-stable.

L-stable methods are very successful in the integration of stiff systems. They are especially suited to dissipative systems, since they closely capture the rapid dissipation of the energy held in the fast modes. However, care should be taken when applying an L-stable method to other systems, where the fastest modes may be of importance, for instance systems with highly oscillatory solutions where some large-magnitude eigenvalues are on the imaginary axis. In that case, the numerical integration with an L-stable method may unrealistically damp these modes if the time step is too large.

Some sources of stiffness for PDEs include: strongly refined space discretisations for diffusive processes, presence of strong spatial gradients in the solution. Sources of stiffness which are relatively independent of the space discretisation include rapid chemical reactions, and more generally processes with vastly different time scales. A usually very informative picture of the stiffness of an ODE system  $y' = f(y)$  can be obtained by looking at the distribution of the eigenvalues of the Jacobian  $d_y f$ . If they are clustered near 0, an explicit method will most likely be very appropriate. If on the other hand some eigenvalues have a very large negative real part, L-stable methods will be much more efficient.

L-stability and stiff accuracy are also very important properties for the integration of DAEs, as seen in the next section.

## 5.1.2 Convergence of Runge-Kutta methods for index-1 DAEs

In Chapter 4, we have seen that the system arising from the space discretisation of the unsteady model from Chapter 2 is a DAE system of index 1. In this section, we briefly review the main theorems regarding the application of Runge-Kutta methods to DAEs of index 1. More details are available in the classical books [206, 218].

## Runge-Kutta methods applied to the semi-explicit form of the DAE

Runge-Kutta methods were originally designed for ODE system of the form  $y' = f(y)$ . Thus, we can straightforwardly apply such a scheme to the differential part of the semi-explicit DAE (4.24) and (4.25), but not to the constraints. To circumvent this issue, it is useful to reformulate the DAE as the singular perturbation limit of a problem involving a small parameter  $\epsilon$ :

$$d_t W = f(W, Z) \quad (5.18)$$

$$\epsilon d_t Z = g(W, Z) \quad (5.19)$$

i.e. we reintroduce a temporal derivative for the algebraic variables  $Z$ , with the aim of inducing a rapid return of  $Z$  onto the set  $\{Z \in \mathbb{R}^n; g(W, Z) = 0\}$ . Any nonzero value of  $g$  encountered at one point in time will lead to a rapid transient of characteristic time  $1/\epsilon$  during which the value of  $Z$  will change to return on the previous set. As we lower  $\epsilon$ , the transient becomes faster, and the stiffness of the overall system increases as the time scales of  $W$  remain approximately unaffected. When  $\epsilon \rightarrow 0$ , it is expected that  $g(W, Z) = 0$  at every point in time.

Note that depending on  $g$ , the dynamics of  $z$  may be divergent. For instance, if  $g(W, Z) = W - Z$ ,  $Z$  will evolve towards  $W$ . If we change the sign of  $g$ , e.g.  $g(W, Z) = -W + Z$ , then Equation (5.19) will let  $Z$  diverge. Thus a more generic choice [218] is to state  $-\epsilon(\partial_Z g)d_t z = g(W, Z)$ . This can be seen as a continuous Newton step on the nonlinear problem  $g = 0$ , accelerated by a factor  $1/\epsilon$ . Note that with this formulation, if  $(\partial_Z g)$  is singular, the previous singular perturbation problem is itself a DAE...

A generic s-stage Runge-Kutta method applied to the singular perturbation problem reads:

$$\left\{ \begin{array}{l} w_{ni} = W_n + \Delta t \sum_{j=1}^s a_{ij} f(w_{nj}, z_{nj}) \quad (5.20) \\ \epsilon z_{ni} = \epsilon Z_n + \Delta t \sum_{j=1}^s a_{ij} g(w_{nj}, z_{nj}) \quad (5.21) \\ W_{n+1} = W_n + \Delta t \sum_{i=1}^s b_i f(w_{ni}, z_{ni}) \quad (5.22) \\ \epsilon Z_{n+1} = \epsilon Z_n + \Delta t \sum_{i=1}^s b_i g(w_{ni}, z_{ni}) \quad (5.23) \end{array} \right.$$

where  $w_{ni}$  and  $z_{ni}$  represent the values of  $W$  and  $Z$  at stage  $i$ , and  $a_{ij}$ ,  $b_i$  and  $c_i$  are the coefficients of the Runge-Kutta method.

If we directly let  $\epsilon = 0$  to recover the DAE case, the value of  $Z_{n+1}$  disappears. To circumvent that issue, we first reformulate Equation (5.23) to let  $\epsilon$  appear in the sum in the right-hand side. Let us assume the Runge-Kutta method is implicit and its Butcher matrix  $A = (a_{ij})$  invertible, and let  $\omega_{ij}$  the components of the inverse of  $A$ . We can rewrite Equation (5.21) as:

$$\Delta t g(w_{ni}, z_{ni}) = \epsilon \sum_{j=1}^s \omega_{ij} (z_{ni} - Z_n) \quad (5.24)$$

Reinjecting this expression in Equation (5.23) yields:

$$\epsilon Z_{n+1} = \epsilon Z_n + \epsilon \sum_{i,j=1}^s b_i \omega_{ij} (z_{ni} - Z_n) \quad (5.25)$$

from which  $\epsilon$  can now be eliminated.

To recover our original DAE, we now let  $\epsilon = 0$  in the above formulae:

$$\left\{ \begin{array}{l} w_{ni} = W_n + \Delta t \sum_{j=1}^s a_{ij} f(w_{nj}, z_{nj}) \end{array} \right. \quad (5.26)$$

$$0 = g(w_{ni}, z_{ni}) \quad (5.27)$$

$$\left\{ \begin{array}{l} W_{n+1} = w_n + \Delta t \sum_{i=1}^s b_i f(w_{ni}, z_{ni}) \end{array} \right. \quad (5.28)$$

$$\left\{ \begin{array}{l} Z_{n+1} = (1 - \sum_{i,j=1}^s b_i \omega_{ij}) Z_n + \sum_{i,j=1}^s b_i \omega_{ij} z_{nj} \end{array} \right. \quad (5.29)$$

The same system is obtained in [206, page 375]. Equation (5.27) indicates that, at each stage, the algebraic variables gathered in  $Z$  are determined such that the constraints are all verified. This can be interpreted as a systematic projection of the algebraic variables onto the set  $\{Z; g(W, Z) = 0\}$ . After all stages are computed, the advancement to the next time step is performed via Equations (5.28) and (5.29).

An issue arises from Equation (5.29) which does not necessarily ensure that  $g(W_{n+1}, Z_{n+1}) = 0$ , hence a deviation from the correct solution may occur. A first idea could be to simply replace Equation (5.29) by:

$$0 = g(W_{n+1}, Z_{n+1}) \quad (5.30)$$

to ensure the solution does not deviate from the manifold  $g(y, z) = 0$ . However, such a projection process makes the numerical solution process more complex, because the nonlinear system solved for this projection is different from the system solved for the internal stages. An alternative is to consider stiffly accurate Runge-Kutta methods that satisfy Definition 5.1.3. Then  $W_{n+1} = w_{ns}$ ,  $Z_{n+1} = z_{ns}$ , i.e. the last internal stage is the solution at the next time step. With such methods, we directly obtain  $0 = g(W_{n+1}, Z_{n+1})$ .

In that case, all the internal stages and the final stage of each Runge-Kutta step satisfy  $g(W, Z) = 0$ . As we know that  $(\partial_Z g)$  is invertible, the implicit function theorem indicates that the problem  $g(W, Z) = 0$  locally possesses a unique solution  $Z = G(W)$ , such that the DAE system (4.24) and (4.25) is equivalent to the following ODE:

$$d_t W = f(W, G(W)) \quad (5.31)$$

Applying a stiffly accurate Runge-Kutta methods to the DAE system is therefore equivalent to applying it to the equivalent ODE<sup>1</sup>. Thus the convergence of such methods is the same as for the ODE case, in particular, a stiffly accurate method of order  $p$ , yields the following local truncation at the end of each step:

$$W_{n+1} = W(t_{n+1}) + \mathcal{O}(\Delta t^{p+1}) \quad (5.32)$$

$$Z_{n+1} = Z(t_{n+1}) + \mathcal{O}(\Delta t^{p+1}) \quad (5.33)$$

Non-stiffly accurate methods have their order of convergence severely reduced on the algebraic variables, and methods with  $|R(\infty)| > 1$  are unstable [206, page 380].

---

<sup>1</sup>Note that, in a practical situation, the function  $G$  is most likely unknown. The values  $Z = G(W)$  are thus constructed iteratively, for instance by applying a Newton method on  $0 = g(W, Z)$ , with a sufficiently stringent convergence criterion.

Interpreting the DAE system (4.24)-(4.25) as the limit when  $\epsilon \rightarrow 0$  of the singular perturbation problem (5.18)-(5.19) shows that the algebraic nature of DAEs is related to the presence of infinitely fast modes. Hence it is sensible that L-stability and stiff accuracy are important properties.

Finally, the problem we aim at describing (unsteady model from Chapter 2) is stiff for multiple reasons, among which are the presence of rapid chemical reactions, and the finely resolved heat diffusion within the solid phase. Therefore, to ensure proper convergence of the numerical solution, L-stability is a very advantageous property. It ensures that modes with time scales much shorter than the time step are instantaneously relaxed.

Some methods presented further down, known as ESDIRK, have a non-invertible Butcher matrix  $A$ , hence some of the previous results are not applicable directly. The theory of stability for DAEs and ODEs has however been extended to such a case [228].

## Examples

The most widely used implicit method is implicit Euler (or backward Euler), which is first-order accurate, stiffly accurate and L-stable. If we apply it to the semi-explicit DAE form of our problem, we obtain:

$$\begin{cases} W_{n+1} = W_n + \Delta t f(W_{n+1}, Z_{n+1}) & (5.34) \\ 0 = g(W_{n+1}, Z_{n+1}) & (5.35) \end{cases}$$

A classical second-order scheme is the Crank-Nicolson method [226], which yields:

$$\begin{cases} W_{n+1} = W_n + \frac{\Delta t}{2} (f(W_{n+1}, Z_{n+1}) + f(W_n, Z_n)) & (5.36) \\ g(W_{n+1}, Z_{n+1}) = -g(W_n, Z_n) & (5.37) \end{cases}$$

Equation (5.37) may be surprising, however it shows that this method is sensitive to error accumulation on the algebraic variables. In particular, it is crucial that the initial condition satisfies  $g(Y_0, Z_0) = 0$ . This method is not L-stable and may encounter difficulties when applied to stiff systems.

### 5.1.3 Optimising the computational cost

When advancing forward in time, Equations (5.26) and (5.27) must be solved, usually via a Newton algorithm, which iterates on the values  $w_{ni}$  and  $z_{ni}$  for  $i \in [1, s]$ . Fully implicit Runge-Kutta methods are such that all the stages must be solved simultaneously. The very popular stiffly accurate method Radau5 [206], a 3-stage fifth-order fully implicit method based on Gauss-Radau quadrature points, is one such method. It possesses very interesting properties, however if the problem has  $N$  unknowns, each time step requires solving a  $3N \times 3N$  system, which can be rather costly. An appealing subclass of Runge-Kutta methods is the class of diagonally-implicit Runge-Kutta methods (DIRK) [229]. These methods are such that the summations in Equations (5.26) and (5.27) for the  $i$ -th stage only go up to  $i$  instead of  $s$ , i.e. any stage can be solved by knowing the values at the previous stages. Such methods require more stages (typically twice as many) to reach the same order of convergence as fully implicit methods, however a complete time step only requires the resolution of  $s$  systems of size  $N \times N$ , which is usually more computationally efficient. Therefore we narrow down our choices to DIRK methods.

### 5.1.4 Newton algorithm

For DIRK methods, the stages can be computed sequentially. For each stage we need to solve Equations (5.26) and (5.27) for the unknowns  $w_{ni}$  and  $z_{ni}$ . These equations can be combined to form the nonlinear problem on the values  $(w, z)$  of the differential and algebraic variables at this internal stage:

$$F(w, z) = \begin{pmatrix} w - a_{ii}\Delta t f(w, z) - R \\ g(w, z) \end{pmatrix} = 0 \quad (5.38)$$

with  $R = \Delta t \sum_{j=1}^{i-1} a_{ij} f(w_{nj}, w_{nj})$  gathering the effects of the previous stages, without any dependence on  $(w, z)$ .

Introducing  $X = (w, z)^t$  the vector of unknowns, we can reformulate this problem in the more compact form  $F(X) = 0$ . This problem is solved iteratively using a modified Newton-Raphson method with damping, simply referred to as Newton method in the rest of this work. Some technical details are given in Section 10.1.2. Starting from an initial guess, the method generates a sequence of iterates  $(X_\alpha)$  such that:

$$X_{\alpha+1} = X_\alpha - \tau_\alpha J^{-1} F(X_\alpha) \quad (5.39)$$

The Jacobian  $J = \partial_X F$  is only updated when the convergence is poor or if the iterates diverge. It is usually computed by finite differences. The damping coefficient  $\tau_\alpha$  is initially set to 1. It is reduced as long as the norm of the Newton step is not decreasing, i.e. if  $\|J^{-1}F(X_{\alpha+1})\| > \|J^{-1}F(X_\alpha)\|$ . An error is raised if the Jacobian is computed more than a maximum allowed number of times (typically 5), or if the number of iterations is too high.

The Newton increment  $\Delta X_\alpha = J^{-1}F(X_\alpha)$  is obtained by solving the linear system  $J\Delta X_\alpha = F(X_\alpha)$ . As explained in Section 4.2 for the finite-volume space discretisation of our 1D propellant model, the computation of the interface fluxes only relies on two adjacent cells, therefore  $J$  is block-tridiagonal. A Thomas algorithm is used to solve this system, after having performed a block LU-decomposition.

For our problem, the Jacobian of the nonlinear problem can be expressed in terms of the differential and algebraic variables:

$$\frac{\partial F}{\partial(w, z)} = \begin{pmatrix} \text{I} - \Delta t a_{ii} \partial_w f & -\Delta t a_{ii} \partial_z f \\ \partial_w g & \partial_z g \end{pmatrix} \quad (5.40)$$

If the original system is not too nonlinear, it is highly likely that the Jacobians of  $f$  and  $g$  will not vary much as  $w$  and  $z$  changes from one iteration to the next. However, if the method is composed of multiple stages and the coefficients  $a_{ii}$  are not equal, the Newton algorithm will need to update the Jacobian at each stage. Being able to reuse the Jacobian as many times as possible is important to save computational time, therefore we focus on singly-diagonally implicit Runge-Kutta methods (SDIRK), which satisfy the property  $a_{ii} = a_{jj} \forall (i, j) \in \llbracket 1, s \rrbracket$ .

### 5.1.5 Time adaptation

#### Motivation

A typical ignition transient is shown in Figure 6.4a for a solid propellant ignited by a laser source. The evolution of the surface temperature is very rapid at the beginning of the propellant heating,

before slowing down. At the time of ignition, the temperature rises again quickly before settling to its steady-state value. The ignition transient can be split into successive phases with very different time scales. Consequently, ensuring the time step is adapted to the time scale of each phase is very important to guarantee a precise resolution and to save computational time.

Some simulations presented in the literature use a constant time step, taken as sufficiently low compared to the ignition time [32, 36]. When time adaptation is used, it usually relies on a CFL criterion [33]. CFL limitation may be irrelevant before ignition, as the gas-phase flow velocity is negligible, therefore it can be supplemented with an additional requirement that the relative solution variation between two successive time steps is sufficiently small (e.g. 1%). However such an approach requires fine-tuning and does not provide any guarantee regarding the accuracy of the solution. Embedded Runge-Kutta methods provide a local error estimate by comparing two solutions at different orders, allowing for a precision-driven time step to be computed, which truly reflects the accuracy of the numerical integration. The process is described in [224, page 165] and is briefly recalled here for completeness.

### Embedded methods

An embedded Runge-Kutta method provides two approximations of different orders for the solution. Let us consider a generic scalar ODE  $y' = f(y)$ , with the Runge-Kutta method providing a first approximation  $y_{n+1}$  of order  $p$ , and a second one  $\hat{y}_{n+1}$  of order  $q < p$ . Let us denote as  $y(t)$  the exact solution. Assuming we start with the exact solution  $y(t_n)$  as initial condition for the time step  $n$ , the local errors for the two Runge-Kutta solutions are:  $y_{n+1} = y(t_{n+1}) + \mathcal{O}(\Delta t^{p+1})$  and  $\hat{y}_{n+1} = y(t_{n+1}) + \mathcal{O}(\Delta t^{q+1})$ . Therefore the difference is  $\epsilon(\Delta t) = |y_{n+1} - \hat{y}_{n+1}| = |\mathcal{O}(\Delta t^{p+1}) - \mathcal{O}(\Delta t^{q+1})| \approx \mathcal{O}(\Delta t^{q+1}) = \alpha \Delta t^{q+1}$ , with  $\alpha > 0$  a constant. We can assume the time evolution of the solution is sufficiently well resolved when this difference is smaller than a specified tolerance:  $\epsilon(\Delta t) \leq tol$ . Defining the integration error as  $err(\Delta t) = \epsilon(\Delta t)/tol$ , this is equivalent to  $err(\Delta t) \leq 1$ . If we choose a time step  $\Delta t_1$  such that this condition is not met, we can estimate a time step  $\Delta t_{opt}$  such that the error matches the tolerance:

$$\epsilon(\Delta t_1) = \alpha \Delta t_1^{q+1} > tol, \quad \epsilon(\Delta t_{opt}) = \alpha \Delta t_{opt}^{q+1} = tol$$

Dividing the second equation by the first one yields:  $(\Delta t_{opt}/\Delta t_1)^{q+1} = tol/\epsilon(\Delta t_1)$ , therefore:

$$\Delta t_{opt} = \Delta t_1 (tol/\epsilon(\Delta t_1))^{1/(q+1)} \quad (5.41)$$

If  $\Delta t_1$  is such that the asymptotic regime of convergence for the integration method is already reached, then the estimated local integration error for  $\Delta t = \Delta t_{opt}$  will be close to the prescribed tolerance. If the estimated error is larger than the tolerance, the current step is restarted with the new (smaller) time step  $\Delta t_{opt}$ . Otherwise, it is accepted and the next step is computed with the new optimal time step length  $\Delta t_{opt}$ . This allows the time step to be dynamically reduced or increased, ensuring that the solution is solved at least as precisely as specified, while minimising computational cost. In practice, to avoid over-correcting the time step, we do not allow it to change by a factor lower than 0.2 or higher than 5 between two successive error estimations. A safety factor of 0.9 is also applied to  $\Delta t_{opt}$  to ensure the tolerance is strictly satisfied. If a time step fails due to floating arithmetic errors or results in non-convergence of the Newton iterates, the current step is started over again with a decreased step length. If the time step becomes smaller than  $10^{-15}$  s, an error is raised to avoid important numerical rounding errors.

In the case of ODEs, the estimated integration error is defined as in [224, page 168]. Relative tolerances  $rtol$  and absolute tolerances  $atol$  are defined as scalars and a tolerance vector is defined as  $tol = atole + rtol|y_n|$ ,  $y_n$  the solution at the former time step and  $\mathbf{e}$  the vector of ones (same size as  $y_n$ ). The integration error is then estimated as:

$$err_{n+1} = \left\| \frac{y_{n+1} - \hat{y}_{n+1}}{tol} \right\|_2 \quad (5.42)$$

where the fraction stands for division element by element. In this manuscript we use the 2-norm.

Two potential pitfalls should be stressed for the integration of stiff ODE systems. The first one is the potential lack of L-stability for the error estimate. This can occur if the main method is L-stable and the embedded method is not. In that case, fast error modes may dominate the estimated error, leading to an overestimation of the actual true error. To circumvent that issue, a pragmatic and efficient idea has been proposed by Shampine [230]. For an ODE system  $d_t y = f(y)$ , it consists in computing a corrected error estimate  $\widetilde{err}$  by solving the following linear system:

$$(\mathbf{I} - \Delta t a_{ii} \partial_y f) \widetilde{err}_{n+1} = err_{n+1} \quad (5.43)$$

For simplicity, let us assume that  $\partial_y f$  is diagonalisable with eigenvalues  $\lambda_j$ . We see that for slow modes ( $|\lambda_j \Delta t| \ll 1$ ), the corrected error estimate is equal to the original estimate. For fast modes however ( $|\lambda_j \Delta t| \gg 1$ ), the error components are divided by  $|\lambda_j \Delta t|$ , i.e. they are greatly reduced. This strategy has been shown to be very useful in many ODE solvers [206, page 123]. It can be interpreted as performing a single implicit Euler step on the linear ODE  $\widetilde{err} = (\partial_y f) \widetilde{err}$ , with initial condition  $\widetilde{err}(0) = err_{n+1}$ , and a time step of length  $\Delta t a_{ii}$ . Since implicit Euler is L-stable, this procedure effectively “L-stabilises” the error estimate. This may reduce the number of integration steps taken and the number of rejected steps. It is also very convenient to use, since the matrix  $(\mathbf{I} - \Delta t a_{ii} \partial_y f)$  is already available and factorised in the Newton procedure.

For the present work, this estimate may be adapted to the framework of DAEs. However, we have observed that the uncorrected estimate performed well, thus we did not include this corrective strategy.

The second pitfall is a potential lack of step size stability [231, 232]. The origin of the problem is the following. The optimal time step  $\Delta t_{opt,n+1}$  is computed such that the estimated error matches the tolerance during the integration step from  $t_n$  to  $t_{n+1}$ . Once this step has been accepted,  $\Delta t_{opt,n+1}$  is used as the initial time step choice for the integration step from  $t_{n+1}$  to  $t_{n+2}$ . This is only a sensible approach if the error constant  $\alpha$  remains unchanged. In practice however, this term depends on the solution and may vary rapidly from one step to the next [231]. If  $\alpha$  greatly increases, using  $\Delta t_{opt,n+1}$  for the step from  $t_{n+1}$  to  $t_{n+2}$  will result in an estimated local error  $err_{n+2}$  that is larger than the tolerance, hence the time step will be rejected. This can occur repeatedly, hurting the efficiency of the method. If  $\alpha$  decreases, the step will be accepted but the error will be too low compared to the tolerance, hence the efficiency of the overall integration will not be optimal.

To improve on that, advanced time step controllers have been devised, with the first ones developed by Gustafsson [231], which rely on the values of  $err$  and  $\Delta t$  at the last two previous steps ( $n$  and  $n - 1$ ) to predict the evolution of  $\alpha$  for the next step. These are intimately related to digital filters. Since we have not experienced extensive step rejection in our simulations, we do not investigate this strategy further.

### 5.1.6 Final choice of the method

Considerations on the accuracy of the method for a system of DAE of index 1 has led us to consider stiffly accurate Runge-Kutta methods. Minimisation of the computational cost due to Newton iterations is ensured by using singly-diagonally implicit methods. Finally, the requirement of native time adaptation capabilities favours embedded methods. Overall, based on the criteria presented so far, we look for embedded stiffly accurate SDIRK methods. Additionally, improvements in the error estimation for stiff system and DAEs can be obtained if the lower-order embedded method is stiffly accurate as well, as discussed in [68]. This reference introduces several such schemes with an additional interesting property, which is that the first stage values are equal to the former time step values. This allows to have a “free” stage to improve the accuracy of the method without any additional cost. We retain three schemes from this reference:

- ESDIRK-32A, a four-stage, third-order method with a second-order embedded scheme
- ESDIRK-43B, a five-stage, fourth-order method with a third-order embedded scheme
- ESDIRK-54A, a seven-stage, fifth-order method with a fourth-order embedded scheme

The coefficients of ESDIRK-54A are given in Table 5.1. Other methods may be applicable. Rosenbrock and multistep methods can be applied directly to the semi-explicit form of the system. In particular the DASSL algorithm [218] has been extensively applied to DAE systems of index 1 with great success and may be an interesting alternative to our approach. Another possibility is to take advantage of the fact that the DAE system is of index 1, thus the algebraic variables can be uniquely determined from the differential variables, i.e. we can consider that there exists a function  $p$  such that  $Z = p(W)$  satisfies  $g(W, Z) = 0$ . Usually this relation is not explicitly known, therefore  $Z$  is determined iteratively via a Newton method. The differential variables  $W$  are then governed by the differential equation  $d_t W = f(W, p(W))$ , which can be integrated with any ODE solver, in particular explicit ones. However we have found that our fully implicit approach is able to produce accurate results while having CFL numbers much higher than 1, therefore explicit integration algorithms would be relatively inefficient due to stability requirements. A partially implicit algorithm, e.g. IMEX methods [233], could be used to remove such numerical instabilities induced by the convection operator, however diffusion and reaction operators would also cause stability issues if treated explicitly, thus defeating the purpose of IMEX methods. Splitting methods could be used so that each phenomenon (diffusion, convection, reaction) is integrated with an adequate efficient method, however the order of accuracy in time would generally not exceed 2. Furthermore, additional difficulties may appear when handling the algebraic constraints, and time step adaptation is more involved compared to embedded methods [234, 235].

From a practical point of view, if an existing code uses an implicit Euler or Crank-Nicolson scheme, as is often the case in the literature, the implementation of the above strategy is straightforward, as it only requires solving more stages, which are all equivalent to an implicit Euler step, in terms of the nonlinear system to be solved. Only the right-hand side is different, as it involves contributions from the previous stages. This is also the case if the original code uses a multistep BDF method. Once the ESDIRK schemes are properly implemented, adding the time adaptation algorithm is also a straightforward operation.

## 5.2 Handling the continuity equation

As explained in Section 4.3, the discretised mass flow rate field  $m_i = \rho u_i$  is a vector of algebraic variables. The associated constraint is the semi-discrete continuity equation in the gas phase (4.6). Here, we show how the Runge-Kutta formulation can be used in order to provide an original treatment of the mass flow rate constraint in the gas phase. Let us recall that in our one-dimensional low-Mach approach, the density  $\rho$  cannot be considered a true variable of our problem, as it is uniquely determined from the temperature, mass fractions and pressure. Consequently, the continuity equation (4.6) should not be considered as an ODE on  $\rho$ , but rather as a constraint on the mass flow rate field coming from its PDE counterpart:  $\partial_x m = -\partial_t \rho$ . This means that the mass flow rate field  $m$  is governed by a first-order equation in space, with a source term that is the temporal variation of the density. There are at least two different ways of treating this equation, which we describe in this subsection.

### General formulation of our DAE problem

The DAE problem (coming from the semi-discretisation in space) we seek to solve has the form:

$$\begin{cases} \partial_t y = f(y, z_1, z_2) & (5.44) \\ 0 = \partial_t \phi(y) + g_1(y, z_1, z_2) & (5.45) \\ 0 = g_2(y, z_2) & (5.46) \end{cases}$$

with index-1 constraints. In our case,  $y = (T, Y_k)$ ,  $z_1 = (\rho u)$ ,  $z_2 = (T_s, Y_{s,k})$ , and  $\phi(y_1) = \rho$  as per the law of state. The constraint (5.45) represents the continuity equation, with  $\phi$  the gas density computed from the ideal gas law (2.19). The constraint (5.46) represents the interface coupling conditions. Equation (5.44) gathers the remaining differential equations on  $\rho h$ ,  $\rho Y_k$ . Multiple time discretisation approaches can be envisioned and are described next.

### Instantaneous reformulation of the first constraint

Equation (5.45) can be transformed so that the time derivative of  $\phi$ , which is not a true variable of our system, does not appear. Writing:

$$\partial_t \phi(y) = (\partial_y \phi) d_t y \quad (5.47)$$

Equation (5.45) can be replaced by:

$$0 = (\partial_y \phi) f(y, z_1, z_2) + g_1(y, z_1, z_2) \quad (5.48)$$

The temporal integration process is the following. First, we apply a Runge-Kutta (or multistep) scheme on (5.44) to compute the evolution of  $y$ . At each integration step (actually at each Newton step if an implicit method is used), the algebraic variables  $z_1$  are computed by solving equation (5.48), and  $z_2$  is obtained by solving Equation (5.46). We refer to this reformulation of the continuity equation as the *instantaneous flow rate constraint* to highlight the fact that it only requires the knowledge of the solution profiles at the given time point. This reformulation corresponds to the so-called “state-space form” [206] of an index-1 DAE. In this form, the algebraic equations are formulated such that they can be solved separately from the differential ones to obtain the values of the algebraic variables from the differential ones. With this approach, any integration algorithm

(including explicit ones) could be used to compute the evolution of the differential variables with Equation (5.44). However the cost of solving for the algebraic variables would be prohibitive if the overall solution algorithm remains implicit, as they must be computed at each Newton step. On the other hand, explicit algorithms will impose severe stability constraints on the time step and result in very inefficient simulations. Therefore, we believe it is best to solve for the algebraic and differential variables at the same time, i.e. use a fully coupled implicit approach. This state-space form is often used in the literature for fractional-step approaches, in particular in low-Mach or incompressible flow models to compute the perturbed pressure field by solving a Poisson equation to ensure the velocity field remains divergence-free [236–238].

Let us derive the instantaneous formulation for our gas phase. The term  $-\partial_t \rho$  in the continuity equation  $\partial_x m = -\partial_t \rho$  can be considered as a source term which is a function of  $m$ . It can be obtained by differentiating the logarithm of the equation of state (2.19) with respect to time, as classically done in the combustion community (see for example [214, 236–238]). This yields:

$$\partial_x m = -\rho \left( \frac{\partial_t P}{P} - \frac{\partial_t T}{T} - \frac{\sum_{\mathfrak{k}=1}^{n_e} \frac{\partial_t Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}}{\sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}} \right) \quad (5.49)$$

In the right-hand side,  $\partial_t P$  is an input (e.g. constant pressure, or evolution based on a combustion chamber model). The gas density  $\rho$  is directly given by the equation of state (2.19). The other terms can be constructed based on our original gas phase system:

$$\partial_t Y_{\mathfrak{k}} = \frac{1}{\rho} (\partial_t \rho Y_{\mathfrak{k}} - Y_{\mathfrak{k}} \partial_t \rho) \quad (5.50)$$

where  $\partial_t \rho Y_{\mathfrak{k}}$  is given by Equation (2.10), and  $\partial_t \rho$  is replaced by  $-\partial_x m$  as per the continuity equation (2.9).

The term  $\partial_t T$  is slightly more involved. Let us recall the definition of the enthalpy:

$$h = \sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} \left( h_{\mathfrak{k}}^0 + \int_{T_0}^T c_{p,\mathfrak{k}}(a) da \right) \quad (5.51)$$

We can write  $\partial_t h = \frac{1}{\rho} (\partial_t \rho h - h \partial_t \rho)$ , where  $\partial_t \rho h$  is given by Equation (2.11). Let us express  $\partial_t T$  by differentiating with respect to time the definition of the enthalpy:

$$\partial_t h = \sum_{\mathfrak{k}=1}^{n_e} \partial_t Y_{\mathfrak{k}} \left( h_{\mathfrak{k}}^0 + \int_{T_0}^T c_{p,\mathfrak{k}}(a) da \right) + \sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} c_{p,\mathfrak{k}}(T) \partial_t T \quad (5.52)$$

from which we get:

$$\partial_t T = \frac{\partial_t h - \sum_{\mathfrak{k}=1}^{n_e} \partial_t Y_{\mathfrak{k}} \left( h_{\mathfrak{k}}^0 + \int_{T_0}^T c_{p,\mathfrak{k}}(a) da \right)}{\sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} c_{p,\mathfrak{k}}(T)} \quad (5.53)$$

Overall, we obtain the following constraint:

$$\partial_x m = -\rho \left( \frac{\partial_t P}{P} - \frac{\partial_t h - \sum_{\mathfrak{k}=1}^{n_e} \partial_t Y_{\mathfrak{k}} \left( h_{\mathfrak{k}}^0 + \int_{T_0}^T c_{p,\mathfrak{k}}(a) da \right)}{T \sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} c_{p,\mathfrak{k}}(T)} - \frac{\sum_{\mathfrak{k}=1}^{n_e} \frac{\partial_t Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}}{\sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}} \right) \quad (5.54)$$

In the semi-discrete setting, this yields for the  $i$ -th cell:

$$\frac{m_i - m_{i-1}}{x_i - x_{i-1}} = -\rho_i \left( \frac{\partial_t P}{P} - \frac{\partial_t T_i}{T_i} - \frac{\sum_{\mathfrak{k}=1}^{n_e} \frac{\partial_t Y_{i,\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}}{\sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{i,\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}}} \right) \quad (5.55)$$

with  $\partial_t T_i$  and  $\partial_t Y_{i,\mathfrak{k}}$  computed from Equations (5.53) and (5.50) respectively. Note that these expressions also involve the mass flow rate field  $m$ . The temporal derivatives  $\partial_t \rho_i$ ,  $\partial_t (\rho Y_{\mathfrak{k}})_i$ ,  $\partial_t (\rho h)_i$  are evaluated from the semi-discrete conservation equations (4.6) to (4.8). Equation (5.55) would then replace the continuity equation (4.6) in our DAE system.

Solving Equation (5.55) for the discrete values  $m_i$  can be done by transforming it into a problem of the form  $0 = g(m)$  and solving the latter with a Newton method. This system is linear in  $m$ , thus a single Newton step would suffice. However we can see that the Jacobian  $\partial_m g$  involves the solution profiles  $T$ ,  $Y_{\mathfrak{k}}$ , therefore it will need to be updated often as the solution evolves in time and this may end up being costly. This is in agreement with our computational findings: the instantaneous formulation indeed requires more Jacobian updates than the constraint formulation based on the Runge-Kutta quadrature presented hereafter.

### Use of the Runge-Kutta temporal quadrature

Equation (5.45) constitutes an ODE on  $\phi(y_1)$ , which is not a true variable of our problem, but is computed from the other ones via the equation of state. We can nonetheless apply the Runge-Kutta scheme to this equation. For the  $i$ -th stage of a DIRK method, this leads to:

$$\left\{ \begin{array}{l} Y_{ni} = y_n + \Delta t \sum_{j=1}^i a_{ij} f(Y_{nj}, Z_{1,nj}, Z_{2,nj}) \\ \phi(Y_{ni}) = \phi(y_n) + \Delta t \sum_{j=1}^i a_{ij} g_1(Y_{nj}, Z_{1,nj}, Z_{2,nj}) \\ 0 = g_2(Y_{ni}, Z_{2,ni}) \end{array} \right. \quad (5.56)$$

$$\phi(Y_{ni}) = \phi(y_n) + \Delta t \sum_{j=1}^i a_{ij} g_1(Y_{nj}, Z_{1,nj}, Z_{2,nj}) \quad (5.57)$$

$$0 = g_2(Y_{ni}, Z_{2,ni}) \quad (5.58)$$

Equation (5.57) then constitutes an algebraic constraint on  $Z_{1,nj}$ .

For our particular gas phase problem, applying the Runge-Kutta scheme to the semi-discrete continuity equation (4.6) yields a quadrature formula on  $\int_t \partial_t \rho dt$ . Let  $\rho_n^i$  be the density in the  $i$ -th cell at time step  $n$ , and  $\rho_{n,i}^i$  the same density at the  $i$ -th stage of time step  $n$ . For the  $i$ -th stage of any Runge-Kutta method, we obtain:

$$\rho_{n,i}^i - \rho_n^i = \int_{t_n}^{t_{ni}} (d_t \rho^i) dt \approx \Delta t \sum_{j=1}^s a_{ij} (d_t \rho^i)_{n,j} \quad (5.59)$$

where  $(d_t \rho^i)_{n,j}$  is the time derivative of  $\rho^i$  at time  $t = t_n + c_j \Delta t$  (i.e. at the  $j$ -th stage). Based on the semi-discrete mass conservation equation (4.6), it is equal to the numerical approximation of the mass flow rate spatial gradient at this stage. Equation (5.59) can then be interpreted as a constraint on  $m$ :

$$-\frac{m_{n,i}^i - m_{n,i}^{i-1}}{x^i - x^{i-1}} = \frac{\rho_{n,i}^i - \rho_n^i}{a_{ii} \Delta t} + \sum_{j=1, j \neq i}^s \frac{a_{ij}}{a_{ii}} \frac{m_{n,j}^i - m_{n,j}^{i-1}}{x^i - x^{i-1}} \quad (5.60)$$

Comparing this equation to the discrete continuity equation  $-\frac{m_{n,i}^i - m_{n,i}^{i-1}}{x^i - x^{i-1}} = (d_t \rho^i)_{n,i}$ , we see that the right-hand side is the approximation of the source term  $(d_t \rho^i)_{n,i}$  that is naturally constructed by the Runge-Kutta scheme and which can be entirely expressed in terms of the mass flow rates at various stages. To our knowledge, this approach has not been proposed in the one-dimensional low-Mach literature so far.

## Comparing both approaches

We can interpret the application of the Runge-Kutta scheme to the continuity equation as performing a quadrature on the continuity equation, i.e. approximate the integral in the exact solution:

$$\phi(Y_{nj}) = \phi(y_n) + \int_{t_n}^{t_n + c_i \Delta t} g_1(y(t), z_1(t)) dt$$

The quadrature is the same as the one used for the ODE (5.44) and has the same order of accuracy in time.

On the other hand, the instantaneous reformulation yields:

$$0 = (\partial_y \phi)(Y_{nj}) f(Y_{nj}, Z_{1,nj}, Z_{2,nj}) + g_1(Y_{nj}, Z_{1,nj}, Z_{2,nj}) \quad (5.61)$$

The quadrature performed in Equation (5.57) introduces an error  $\mathcal{O}(\Delta t^{p+1})$  with  $p$  the accuracy order of the stage considered (1 or 2 for the internal stages of ESDIRK methods, and the overall order of the method for the last stage of a stiffly accurate method). An error of the same order is introduced in the instantaneous formulation (5.61) through  $Y_{nj}$  and  $Z_{2,nj}$ .

It was assessed on the test cases of Chapter 6 that both approaches yield virtually identical results, in terms of solution profiles, order of convergence, time step evolution. Still, the instantaneous formulation yields a more nonlinear system because of the term  $\partial_y \phi$ . Therefore the Newton algorithm has more issues converging and we have found that, in many cases, this results in a dramatic increase in the number of Jacobian updates, roughly 30% more in highly dynamic simulations (ignition), and up to a factor 10 in certain situations (near steady-state AP combustion from Section 6.4), and therefore causes a substantial slow-down of the computation. Therefore, all the results presented in this work, in particular in the next chapter, are obtained with the Runge-Kutta approach.

## Partial conclusion

In this chapter, we have shown the various properties of the time integration method that are required for a high-accuracy adaptive simulation of the unsteady model from Chapter 2. The algebraic nature of some equations strongly favours the use of implicit L-stable schemes. Embedded Runge-Kutta methods of this type, with the added propriety of stiff accuracy, are a very attractive choice. The computational cost is also optimised by using singly-diagonally implicit schemes.

A particular feature of the simulated model is the continuity equation, which is an index-1 constraint involving the time derivative of the density that is not a true differential variable. At least two ways of handling this equation have been presented, with the more classical state-space form being inferior in terms of computational performance compared to the newly introduced natural formulation, obtained by applying the Runge-Kutta scheme directly on the continuity equation.

The designed numerical strategy has been implemented in a Fortran90 tool called VULC1D, which is based on earlier tools for the simulation of steady-state solid propellant or aluminium combustion, for example used in [112].



# Chapter 6

## Numerical experiments

### Summary

Various test cases are considered to show the efficiency and accuracy of the numerical strategy presented in the previous chapter. High-order temporal integration and adaptive time-stepping prove to be valuable for the study of nonlinear and transient phenomena.

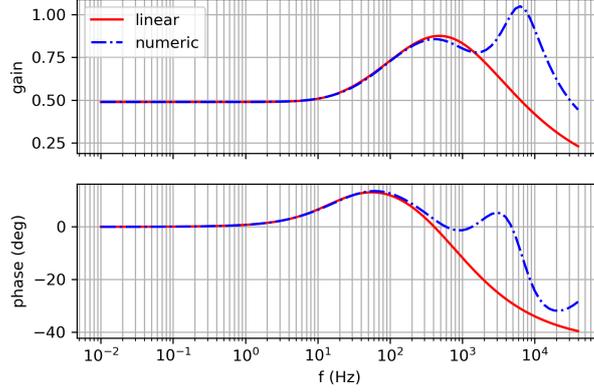
In this chapter, we verify the VULC1D code, developed using the unsteady model from Chapter 2 and the numerical strategy from Chapter 5, by comparing it to other reference solutions obtained theoretically or numerically with other tools.

We then perform extensive numerical tests to demonstrate the adequacy of the numerical strategy for high-accuracy simulations. The test cases include the simulation of laser induced ignition and of the development of a limit cycle for a configuration with a global chemical mechanism in the gas phase, and the simulation of transient combustion for the combustion of AP with detailed and stiff chemistry. The high-order convergence and the time step adaptation capability are shown to be valuable assets to obtain accurate results efficiently.

### 6.1 Verification

The spatial discretisation has already been verified in Section 3.3.3 by comparing the steady-state solution obtained with VULC1D and the solution profile obtained with the semi-analytical tool and enforcing the same level of modelling. We now proceed with the verification of the time integration. No analytical solution is available in the general unsteady regime, however linear frequency responses to pressure oscillations have been available since decades [156]. They constitute a very interesting benchmark, and will be used here as a point of comparison for the response function evaluated numerically with our VULC1D code. We consider the same simple one-dimensional model of an AP-HTPB-Al propellant as in Section 3.3.3, with temperature independent properties in both phases, surface reaction, unitary Lewis number in the gas phase and equal properties (molar mass, heat capacity) for both gaseous species.

The linearised response can be constructed analytically using coefficients that represent the sensitivity of the steady-state solution to certain parameters. These sensitivities can be easily generated with finite differences, using the semi-analytical model or VULC1D. This provides a reference linear response function. The one-dimensional VULC1D code is then initialised with the



**Figure 6.1** Bode diagram of the response function  $R_{mp}$  to pressure fluctuations

steady-state profile and a sinusoidal pressure oscillation  $P(t) = \bar{p} + p' \sin(2\pi ft)$  is forced in the gas phase, with  $f$  a given frequency. The mean pressure  $\bar{P}$  is set to  $= 5 \times 10^6$  Pa, and the amplitude is set to  $p' = 0.001\bar{P}$ . This pressure oscillation leads to fluctuations in the pyrolysis mass flow rate  $m(t) = \bar{m} + m'(t)$ . After a few periods, these oscillations stabilise and we can determine the established response function  $R_{mp} = (m'/\bar{m})/(p'/\bar{p})$  at the corresponding frequency. If we assume that the gas phase is quasi-steady, a linearisation of the heat equation in the solid and of the pyrolysis law yields an expression for the response function [156]:

$$R_{mp}(f) = \frac{nAB}{s + \frac{A}{s} - (1 + A) + AB} \quad (6.1)$$

with  $s = \frac{1}{2} \left( 1 + \sqrt{\frac{1}{2}(y+1)} + i\sqrt{\frac{1}{2}(y-1)} \right)$  and  $y = \sqrt{1 + 16\Omega^2}$ , where  $\Omega = 2\pi f D_c / \bar{r}^2$ ,  $D_c$  is the solid phase thermal diffusivity, and  $\bar{r}$  is the steady-state regression rate. The coefficients  $A$  and  $B$  are defined as:

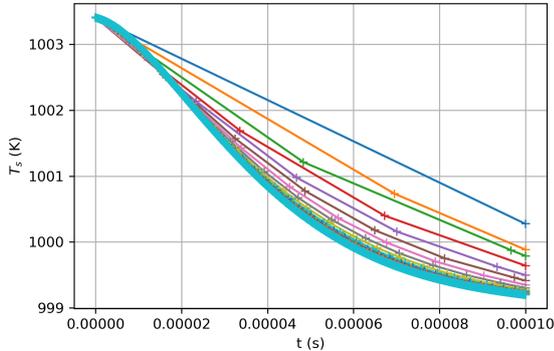
$$A = (\bar{T}_s - T_0) \left( \frac{\partial \ln(\bar{m})}{\partial \bar{T}_s} \right)_{\bar{P}}, \quad B = \frac{1}{(\bar{T}_s - T_0)\sigma_p}, \quad \sigma_p = \left( \frac{\partial \ln(\bar{m})}{\partial T_0} \right)_{\bar{P}} \quad (6.2)$$

Figure 6.1 shows the comparison of the linearised and numerical frequency responses. We see that the agreement between both methods is excellent up to approximately 500 Hz, where the gas phase no longer has a quasi-steady behaviour, thus introducing a larger error in the linear response function. This serves as a global verification of our unsteady model. The secondary peak at high frequencies in the response can be obtained analytically if the unsteady gas phase equations are also linearised, as in [157], however this is a much more involved process.

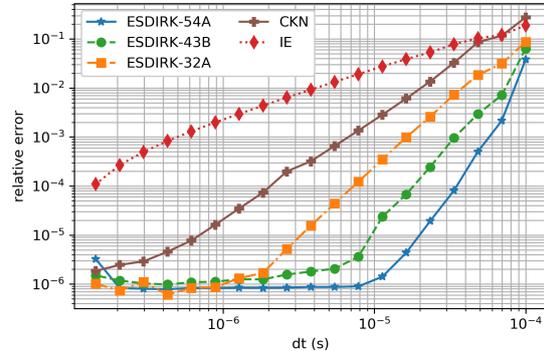
### 6.1.1 Verification of the temporal order of convergence

To verify that a high order of convergence in time can indeed be attained on all variables, a simple test case is set up, using the simplified model presented in Section 3.3.3. A steady-state is computed at pressure  $P = 5.5 \times 10^6$  Pa and is given as initial solution for an unsteady simulation with a lower pressure at  $P = 5 \times 10^6$  Pa. We perform a single step with implicit Euler and  $dt = 10^{-12}$  s to ensure that all constraints are satisfied. The simulation is then run for a physical time of  $10^{-4}$  s with fixed

time steps. The curves of  $T_s$  obtained for various time step values are plotted in Figure 6.2. The cyan curve represents the most refined solution.



**Figure 6.2** Surface temperature histories obtained with IE when gradually lowering the time step



**Figure 6.3** Global error  $\epsilon_{\rho u}$  on the mass flow rate field

A space-averaged relative error is considered for the mass flow rate field at final time ( $N$  global number of cells and  $ref$  for reference simulation):

$$\epsilon_{\rho u} = \sqrt{\sum_0^N \frac{1}{N} \left( \frac{m_i(t_f; dt) - m_i(t_f; dt_{ref})}{m_i(t_f; dt_{ref})} \right)^2} \quad (6.3)$$

This error is plotted in Figure 6.3. The theoretical orders of convergence are attained as long as the error is not limited by the precision of the Newton algorithm and, although not reported here, similar convergence rates are observed for the surface and differential variables (point-wise of space-averaged errors).

Overall, we have shown that the code is second-order accurate in time and up to fifth-order accurate in time. In particular, the presence of algebraic constraints do not hinder the temporal convergence, as expected from the choice of numerical strategy exposed in Chapter 5. The correct implementation of the conservation equations of the unsteady model from Chapter 2 has been verified by comparison to the results of the semi-analytical tool and to the linearised frequency response. We now wish to tackle three much more challenging test-cases and investigate the behaviour of the proposed strategy in terms of accuracy and computational efficiency.

## 6.2 Simulation of a laser-induced ignition transient

To simulate the ignition of a solid propellant, we add a laser heat flux of  $1 \text{ MW}\cdot\text{m}^{-2}$ , which is partially absorbed at the surface, as an additional heat flux in equation (2.27), and partially absorbed in-depth inside the solid as an additional source term in equation (2.7).

### 6.2.1 Setup

We use the same simplified model as previously described in Section 3.3.3. The initial solution is a uniform temperature field at 300 K, with only combustion products in the gas phase, as a simpler

name	number of stages	L-stable	order
ESDIRK-54A	7 (6)	yes	5
ESDIRK-43B	5 (4)	yes	4
ESDIRK-32A	4 (3)	yes	3
CKN (Crank-Nicolson)	2 (1)	no	2
IE (Implicit Euler)	1 (1)	yes	1

**Table 6.1** Selected Runge-Kutta methods. The numbers in brackets correspond to the number of stages actually solved.

alternative to adding nitrogen as initial gas, without much effect on the ignition process itself. The mesh has 99 cells in the solid phase and 291 in the gas phase. The cells are distributed such that the steady-state temperature profile is well resolved.

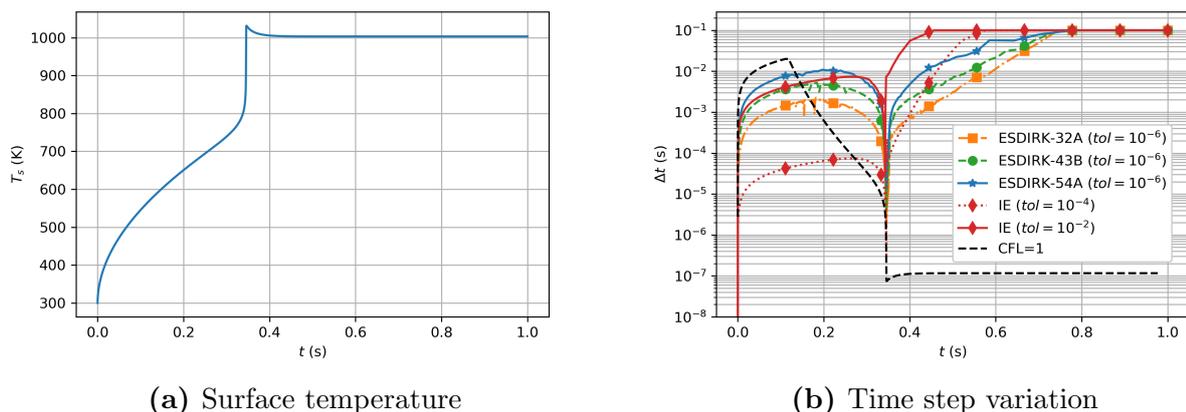
We compute the ignition transient with the methods listed in Table 6.1. The three ESDIRK methods are taken from [68], and they use the time adaptation strategy presented in Section 5.1.5. We also test the classic schemes implicit Euler (IE) and Crank-Nicolson (CKN), with a time adaptation based on the requirement that the solution has a relative variation that is below a certain value between two consecutive time steps. A discussion on a CFL-based time adaptation is presented at the end of this section. We use the abbreviation “tol” to refer to the relative integration error tolerance for ESDIRK methods, and to the allowed relative variation of the solution between consecutive time steps for IE and CKN. The maximum time step allowed is 0.1 s.

Figure 6.4a shows the evolution of the surface temperature during ignition as computed by ESDIRK-54 with  $\text{tol} = 10^{-6}$ . The first phase is the inert heating of the solid propellant. The constant laser heat flux with partial in-depth absorption results in an evolution of  $T_s$  which is very close to being proportional to  $\sqrt{t}$ , that is coherent with the analytical solution of the surface temperature for a solid under a constant surface heat flux.

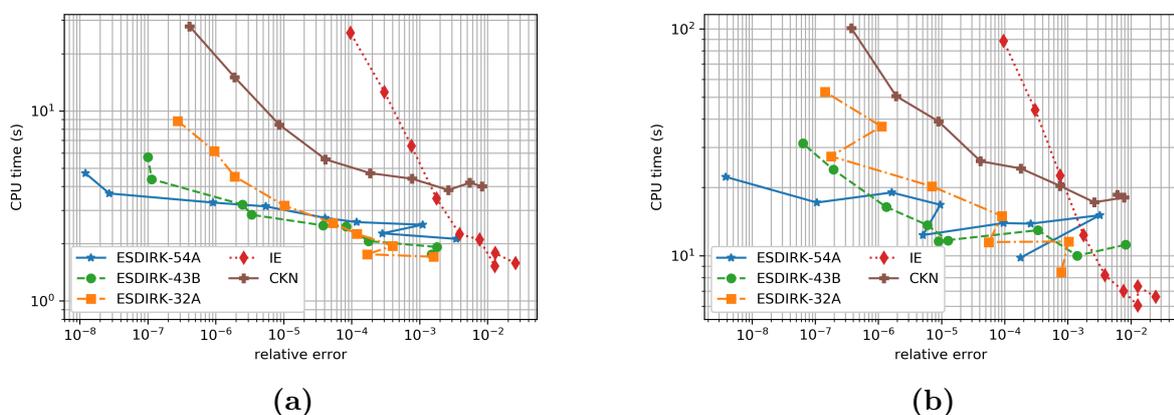
When  $T_s$  is sufficiently high, the pyrolysis mass flow rate given by (2.25) increases rapidly, causing the release of gaseous pyrolysis products in the gas phase, which chemically react and form a flame that heats up the solid even more. Typically at this point, more thermal energy is stored in the solid as compared to steady-state. This results in a momentarily higher regression rate at ignition, seen here near  $t \approx 0.355$  s, which evacuates this excess of solid phase thermal energy. The temperature profile then converges to the augmented steady-state solution, with “augmented” indicating that the laser flux slightly increases the burning rate as compared to the steady-state solution without laser.

## 6.2.2 Results

We compare mainly the evolution of  $T_s$ , the computational time, and the physical time  $t_{ign}$  at which the surface temperature first exceeds 1000 K. Although not shown here, the curves of  $T_s$  for each method are very similar, except for IE simulations with large tolerances that deviate slightly during the inert heating and ignite a few milliseconds earlier. Figure 6.4b shows the evolution of the time step for some of the simulations. We observe that, for ESDIRK embedded methods, increasing the order of the method allows for higher time steps to be used throughout the integration while maintaining the same accuracy. For example, the fifth-order method ESDIRK-54A is capable of taking steps 5 times bigger in average than the third-order method ESDIRK-32A. Finally, it is clear



**Figure 6.4** Ignition transient computed with various methods.



**Figure 6.5** Work-precision diagram for the determination of  $t_{ign}$ : (a) coarse mesh (400 cells), (b) fine mesh (2049 cells).

that IE needs many more steps to achieve a similar result as the ESDIRK methods.

To more quantitatively assess the efficiency of each method in precisely determining the ignition time, multiple simulations were run with each scheme. For the ESDIRK methods, the relative integration error tolerance was varied from  $10^{-1}$  to  $10^{-6}$ . For IE and CKN, the relative solution variation allowed between successive time steps was varied from  $10^{-1}$  to  $10^{-4}$ , without any CFL-limitation. For each simulation the value of  $t_{ign}$  is evaluated and a relative error on this value can be inferred by comparing it to the ignition time obtained with ESDIRK-54A and the tightest tolerance. Figure 6.5a shows the computational time required by each method to achieve a given level of relative error on the initial mesh. Figure 6.5b shows the computational time required by each method on a refined mesh with 734 cells in the solid phase and 1315 cells in the gas phase. Overall in both cases, if a relatively large error on the ignition time, on the order of 1% is deemed sufficient, IE is a relevant choice. If however greater precision is required, ESDIRK schemes with adaptive time stepping as described in Section 5.1.5 are much more efficient. CKN does not perform well for the simulation of an ignition transient.

The black dashed curve in Figure 6.4b shows the time step evolution corresponding to  $CFL = 1$  for the coarse mesh. We clearly see that the CFL-limitation is irrelevant during the inert heating

phase, as the mass flow rate is very low. If, after the inert heating, the CFL is to be limited to low values (1 to 100) as is usually done, many more time steps would be performed. ESDIRK methods are able to give very accurate results without any such limitation. We have determined that, if a simulation was to be performed with a maximum CFL of 10 with ESDIRK-54A and  $rtol = 10^{-6}$  on the physical time interval  $[0, 0.35]$  s, i.e. only up to  $T_s \approx 820$  K before the ignition, the simulation would take 4 times more steps than required without CFL-limitation. If the final physical time is increased to 0.4 s to include most of the transient, the ratio of the number of steps would be 160. Knowing that the simulation without CFL-limitation already achieves an error smaller than  $10^{-6}$  on the ignition time and on the rest of the evolution, this clearly shows that a CFL-constraint is not a good choice in terms of computational efficiency.

The great advantage of embedded methods is that only a relative integration tolerance needs to be specified. No tuning of a CFL-criterion, maximum relative variation or fixed time step is required, hence such methods speed up the engineer task of simulating different scenarios, while still ensuring a controlled error. One interesting observation we made is that the time step values taken by ESDIRK methods were almost identical with both meshes. In all our testing, no correlation was found between the time step evolution of the embedded methods and the mesh refinement. This would not be the case if a CFL-criterion was used. The fully implicit resolution of the coupled solid-gas problem allows to overcome the usual  $CFL = 1$  limitation. This is a strong advantage compared to the methods exposed in [33] for example, where stability issues are reported.

## 6.3 Investigation of limit cycles

The effort made in terms of time integration strategy can be used to accurately study the nonlinear behaviour of the propellant combustion, and in particular potential departures from an unstable steady-state travelling wave solution.

As reported in the literature for other applications [41], high-order methods are often needed to be able to numerically reproduce such a dynamical and nonlinear behaviour, avoiding the excessive numerical damping encountered in classical low-order schemes, e.g. implicit Euler. It is therefore instructive to compare the methods from Table 6.1 already used for the ignition transient in Section 6.2, to see how each of these affects the unsteady result. Namely, it is expected that the integration methods will, depending on their order and the time step used, dampen the oscillating nature of the system and potentially cause a non-physical stabilisation of the solution. Greater reliability is also expected for simulations where dynamic time step adaptation is coupled with high-order accuracy, making it possible to closely follow the behaviour of the system

In this section, we first show how a configuration can be numerically generated, such that the steady-state solution is linearly unstable, but with the extra requirement that initial instability settles on a limit cycle. Then, we choose one such configuration and study the ability of the various temporal schemes considered to capture the initial instability and the limit cycle accurately.

### 6.3.1 Generating configurations with various degrees of instability

In order to highlight the benefit of the high-order adaptive time integration, we search for configurations which are linearly unstable around the corresponding steady-state solution. We generate such configurations with the stable simplified model from Section 3.3.3 as baseline, by varying its parameters. We use existing theoretical tools to approximately evaluate the stability of the steady-state solution.

## Theoretical indicator of intrinsic instability

The Zeldovich-Novozhilov (ZN) framework, already discussed in Section 2.4 is a useful tool to study the stability of a steady-state solid propellant combustion. The core step of the ZN approach is to linearise the heat equation in the solid and the laws (2.58) and (2.59) around the steady-state solution. A mathematical study of the amplification of small compact perturbation leads to the definition of a stability criterion, which depends on two sensitivity coefficients:  $r = (\partial_{T_0} \bar{T}_s)_P$ ,  $k = (\bar{T}_s - T_0) (\partial_{T_0} \ln(\bar{m}))_P$ , with  $\bar{\cdot}$  denoting steady-state values. Steady-state combustion is always stable if  $k < 1$ . If  $k > 1$ , it is stable only if  $r > (k - 1)^2/1 + k$ . The line  $r = (k - 1)^2/1 + k$  is the locus of a Hopf bifurcation, where the steady-state solution becomes linearly unstable in an oscillating manner, with the possibility of stabilising on a limit cycle. If  $r > (\sqrt{k} - 1)^2$ , the instability grows purely exponentially. The associated stability diagram is shown in Figure 6.6a: the leftmost parabola is the first stability limit, the second one is the onset of purely exponential instability. It has been shown that unsteady gas-phase phenomena tend to widen the stability area, however this first simplified analysis remains a good indicator of the stability bounds. We refer the reader to [30] for extensive details on the ZN analysis and its extensions. This stability is called ‘‘intrinsic’’ because it is a property of the solid propellant as an isolated system, as opposed to other types of instabilities, for instance those that might appear when coupling the solid propellant combustion with a chamber [3], whose pressure varies in time based on the propellant regression rate.

## Optimisation problem

We wish to construct unstable variants of the configuration from Section 6.2 with variable degrees of instability, as measured by ZN stability criterion based on the coefficients  $(r, k)$ . In particular, we want to generate configurations with specific positions in the stability diagram, so as to explore the transition between the stability and instability domains.

For that purpose, we setup an optimisation problem. Let us denote as  $X$  the vector containing the parameters of the simple combustion model that we have chosen as free variables. For a given value of  $X$ , we can find the corresponding value of  $(r, k)$  by performing three steady-state simulations with the semi-analytical tool: one baseline simulation, one simulation with a perturbed initial temperature  $T_0$ , and one simulation with a perturbed pressure  $P$ . Then, by means of finite differences,  $r$  and  $k$  may be evaluated. This process can be summarised as the function  $f_{rk} : X \rightarrow (r, k)$ . The optimisation problem is then formulated as:

$$\min_x \quad f_{obj}(x) \quad (6.4a)$$

$$\text{subject to} \quad g(X) \leq 0 \quad (6.4b)$$

$$h(X) = 0 \quad (6.4c)$$

where the objective function  $f$  is defined as  $f_{obj} : X \rightarrow \|f_{rk}(X) - (r, k)_{target}\|_2^2$  with  $(\cdot, \cdot)$  denoting a vector formulation. The inequality constraints are gathered in the vector function  $g$ , and equality constraints are gathered in  $h$ . This problem is simply the constrained minimisation of the distance to the target  $(r, k)$  coefficients. Inequality constraints  $g$  are used to ensure the different physical parameters remain within realistic bounds. They can be supplemented with equality constraints  $h$  to enforce certain properties of the steady-state solution, e.g. surface temperature, regression speed...

Regarding practical implementation, the Sequential Quadratic Programming algorithm SLSQP of the Python library `Scipy` [239] is used. A description of this algorithm will be presented in Chap-

ter 9. The Jacobians of the objective function and constraints are obtained via finite differences. The Hessian of  $f_{obj}$  is built iteratively through a BFGS update. Note that we use the semi-analytical tool to perform the simulation, as it is fast, very precise, does not need any specific mesh generation, and is ensured to converge. For more complex combustion models, the simulations can be carried out with VULC1D, however it is important that the steady-state solutions are converged with sufficient accuracy, so as not to introduce important errors in the Jacobian estimation.

### Numerical assessment of intrinsic stability

We use the previous optimisation problem to generate configurations which have their sensitivity coefficients  $r$  and  $k$  distributed regularly on a segment defined as  $r = 0.137$  (baseline value) and  $k \in [1.5, 1.75]$ , thus crossing the ZN stability limit. The optimisation is constrained as to preserve physically sound characteristics (surface temperature at 1000 K, regression speed of 1 cm/s at 50 atm, 3540 K final flame temperature). The segment and the corresponding targeted points are displayed in Figure 6.6a.

For each point, we numerically assess the stability of the corresponding steady-state combustion. Based on the stationary temperature profile provided by the semi-analytical tool, we generate a non-uniform mesh, such that the increase in temperature between each cell center is a chosen constant  $\Delta T$  (e.g. 10 K). Cells are then added to push the “infinity” boundaries further away, so as to minimise the impact of the Neumann boundary conditions. We typically use  $\Delta T = 20$  K, as we found that the steady-state surface temperature would then match up to less than 1% between the semi-analytical tool and VULC1D. After the mesh has been generated, a slight constant pressure perturbation (typically 0.1% of the prescribed pressure) is applied and the one-dimensional tool is run with implicit Euler and large time steps so as to converge to a perturbed initial steady-state. The pressure is then set back to its original value and an unsteady simulation is performed with ESDIRK-54A and a relative tolerance of  $10^{-6}$  on the integration error. The stability of the combustion can then be assessed numerically by analysing whether the perturbation is damped out or not.

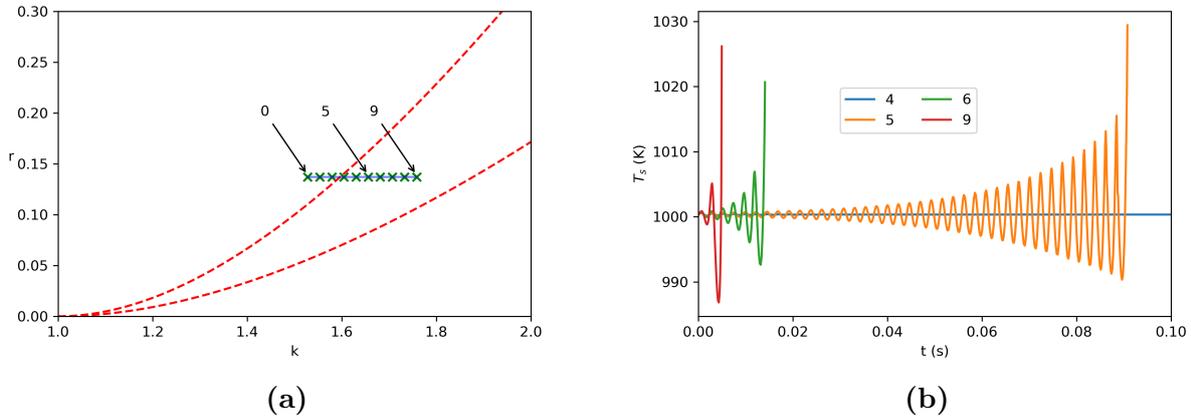
The unsteady simulations for a few points are shown in Figure 6.6b. We see that, up to the 4-th point, the system is stable. Starting from the 5-th point, the system diverges. Overall we observe that the numerical stability limit is slightly further to the right than predicted by the ZN method with quasi-steady gas phase, as already discussed. Between the points 4 and 5 is actually the locus of a Hopf bifurcation, i.e. a point at which the steady-state travelling wave solution becomes unstable. Around the locus of this type of bifurcation, dynamical systems usually exhibit a limit cycle. Iteratively refining the search between points 4 and 5 allows us to find a configuration that produces such a limit cycle.

The corresponding model parameters are the same as in Section 3.3.3, except for the following changes:  $T_{ap} = 14668$  K,  $c_p = 692.8$  J/kg/K,  $c_c = 1253$  J/kg/K,  $T_0 = 182.4$  K,  $\lambda_c = 0.65$  W/m/K,  $\lambda = 0.362$  W/m/K,  $\mathcal{M} = 57.9$  g/mol,  $h_{G_1}^0 = -2.28 \times 10^5$  J/kg,  $h_{G_2}^0 = -2.22 \times 10^6$  J/kg,  $A = 340.4$ .

Note that some properties here are slightly out of the typically expected intervals, e.g  $T_0$  and  $c_p$ . However the goal pursued in this section is not to obtain a perfectly realistic configuration.

The corresponding steady-state temperature profile is shown in Figure 6.7a, as computed by the semi-analytical tool. Using this profile as initial condition and enforcing a small pressure perturbation, an unsteady simulation has been performed with time adaptation, with a relative integration error tolerance  $10^{-6}$ . We obtain the surface temperature evolution plotted in Figure 6.7b. The linear instability causes the system to diverge near  $t = 0$ , however nonlinear effects allow for a stabilisation on a limit cycle after  $t \approx 0.5$  s, where the oscillation amplitude remains constant. The

discrete Fourier Transform of which is presented in Figure 6.7c. The limit cycle oscillations are a sum of sinusoidal harmonic oscillations, with a fundamental frequency close to 452 Hz, which is close to the analytical propellant natural frequencies defined in [156] and [94], at 518 Hz and 348 Hz respectively. This is the configuration we will investigate in the rest of this section.



**Figure 6.6** Generation of unstable configurations: (a) Segment travelled in the  $(r, k)$  stability diagram, (b) Unsteady simulations of some configurations (ESDIRK-54A with  $rtol = 10^{-6}$ )

### 6.3.2 Methodology

The generated unstable configuration is used to study the behaviour of the temporal schemes implemented in VULC1D. First, constant time steps simulations are performed with the ESDIRK methods and other classical schemes. Comparisons are made based on the ability to reproduce the initial amplification of the oscillations, the fundamental frequency of the limit cycle and its amplitude. Second, simulations with adaptive ESDIRK methods are presented. The quality of the results and the computational efficiency are assessed, both with fixed or variable time steps.

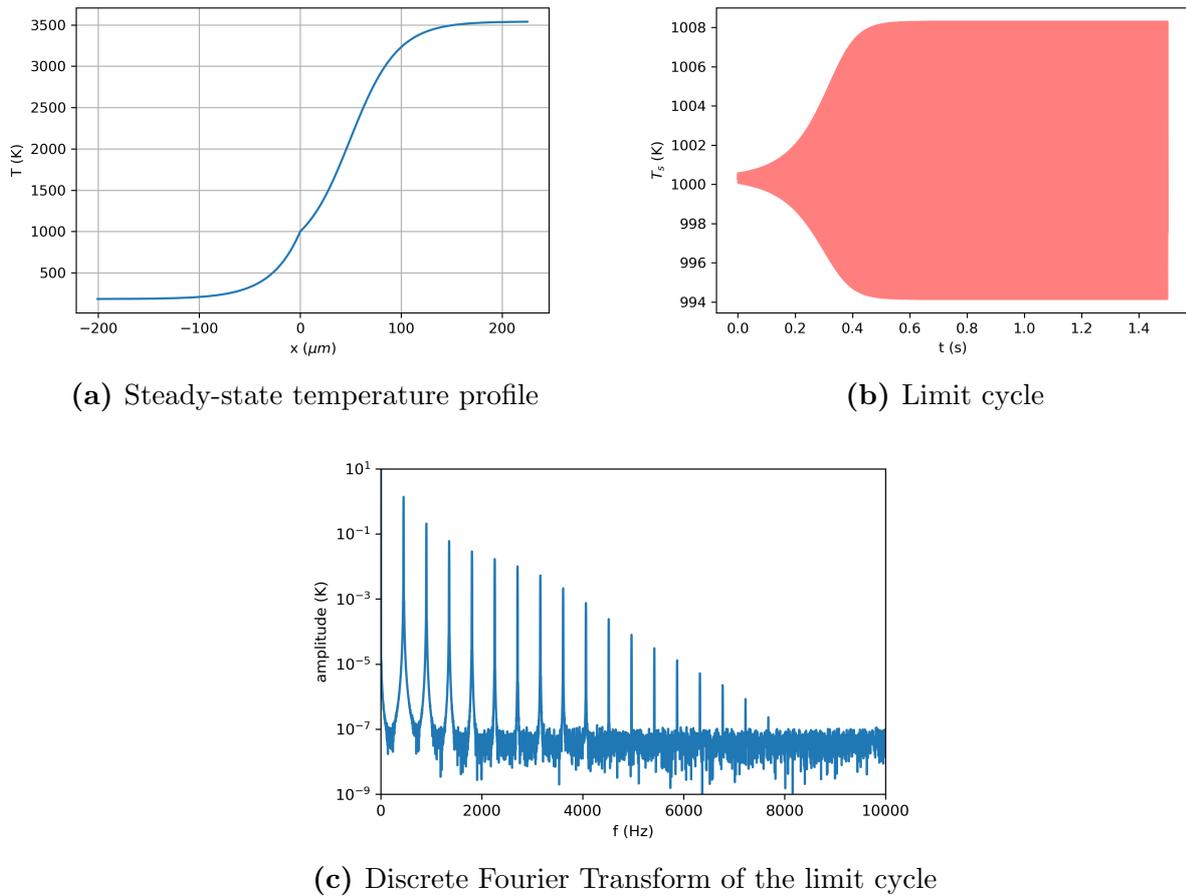
#### Simulation setup

The integrators used are the stiffly accurate Runge-Kutta methods described in Table 6.1. Included are implicit Euler and Crank-Nicolson. The latter has been historically praised for its damping properties. Its absolute stability domain exactly contains all physically stable configurations (in the linear case) and this scheme does not damp purely oscillating linear systems. This method actually is part of the ESDIRK class, as its first stage is explicit. Its main drawback is the lack of L-stability, a property which is very advantageous for index-1 DAE problems.

As described in 6.3.1, the simulations are performed on a non-uniform mesh based on the temperature profile of the steady-state solution with 55 cells for the solid phase and 146 for the gas phase. It has been verified that additional refinement would not affect the solution dynamics.

#### Comparison process

We focus on several aspects. First we analyse visually the envelope curve of the surface temperature time history. For each simulation, this curve is constructed as the junction of the successive maxima of  $T_s$ . Unfortunately, for large values of the time step  $dt$ , the sampling frequency  $1/dt$  can be



**Figure 6.7** Main features of the studied configuration

insufficiently high compared to the fundamental frequency of the limit cycle, causing artefacts to appear in the form of an oscillation of the envelope. This can be improved by using a cubic interpolant of  $T_s$  to determine the successive maxima with greater precision, however envelope oscillations are still present, for example in Figure 6.8e.

Second, we analyse the growth of the linear instability near  $t = 0$ . For each simulation, the best exponential fit for the envelope of the evolution of  $T_s$  is determined, i.e. the curve joining the successive maxima of  $T_s$ , which are obtained from a cubic interpolation of the temporal evolution of  $T_s$ . Such a fit is of the form  $T_{fit}(t) - T_s(0) = A \exp(bt)$ , with  $b$  the fitted amplification factor.

Third, the established limit cycle is considered, on the time window  $1 \leq t \leq 1.5$  s. A discrete Fourier transform of the surface temperature signal is computed via an FFT algorithm, as shown for the reference simulation in Figure 6.7c. Interpolation of the solution on a uniform time grid is performed if the simulation was not conducted with a constant time step. This FFT helps determine the approximate frequencies of the different harmonics with a precision of approximately 1-10 Hz. For each of these peaks in the spectrum, the peak frequency is then precisely computed by maximising the correlation between  $T_s(t)$  and  $\exp(2i\pi ft)$ , from which we can also determine the precise amplitude of the corresponding peak. These values offer a trustworthy and precise indication of how well the limit cycle is captured.

Finally, work-precision diagrams are given which represent the computational time required to achieve a specific level of relative error on the quantitative results, i.e. fitted amplification factor,

fundamental frequency and amplitude. Relative errors are computed relative to the values obtained with the most refined solution.

### 6.3.3 Analysis of schemes efficiency for a constant time step

#### Envelope of the surface temperature history

We first observe the envelope curves of  $T_s$  for various time step values in Figure 6.8. We see that all methods dampen out the oscillations when the time step is too big, except CKN which stabilises at a small oscillating amplitude. However, if we gradually decrease the time step, each method eventually produces a limit cycle. We see that ESDIRK-54A has the best behaviour in terms of reproducing the actual reference limit cycle. It generally seems that the higher the order of the method, the larger the time step can be while still resolving the limit cycle.

An interesting behaviour is observed for the CKN method: though it is second-order accurate, it finds a relatively correct initial amplification with larger time steps than required by the fourth- and third-order methods. However it has the drawback that the solution diverges unless the time step is further reduced. A non-diverging solution with CKN is obtained with time steps for which all higher-order methods already provide better results. It seems that the lack of L-stability favours the initial destabilisation, but leads to the divergence of the solution. On the opposite, all L-stable methods dampen the oscillations when the time step is too big, and none of them diverges.

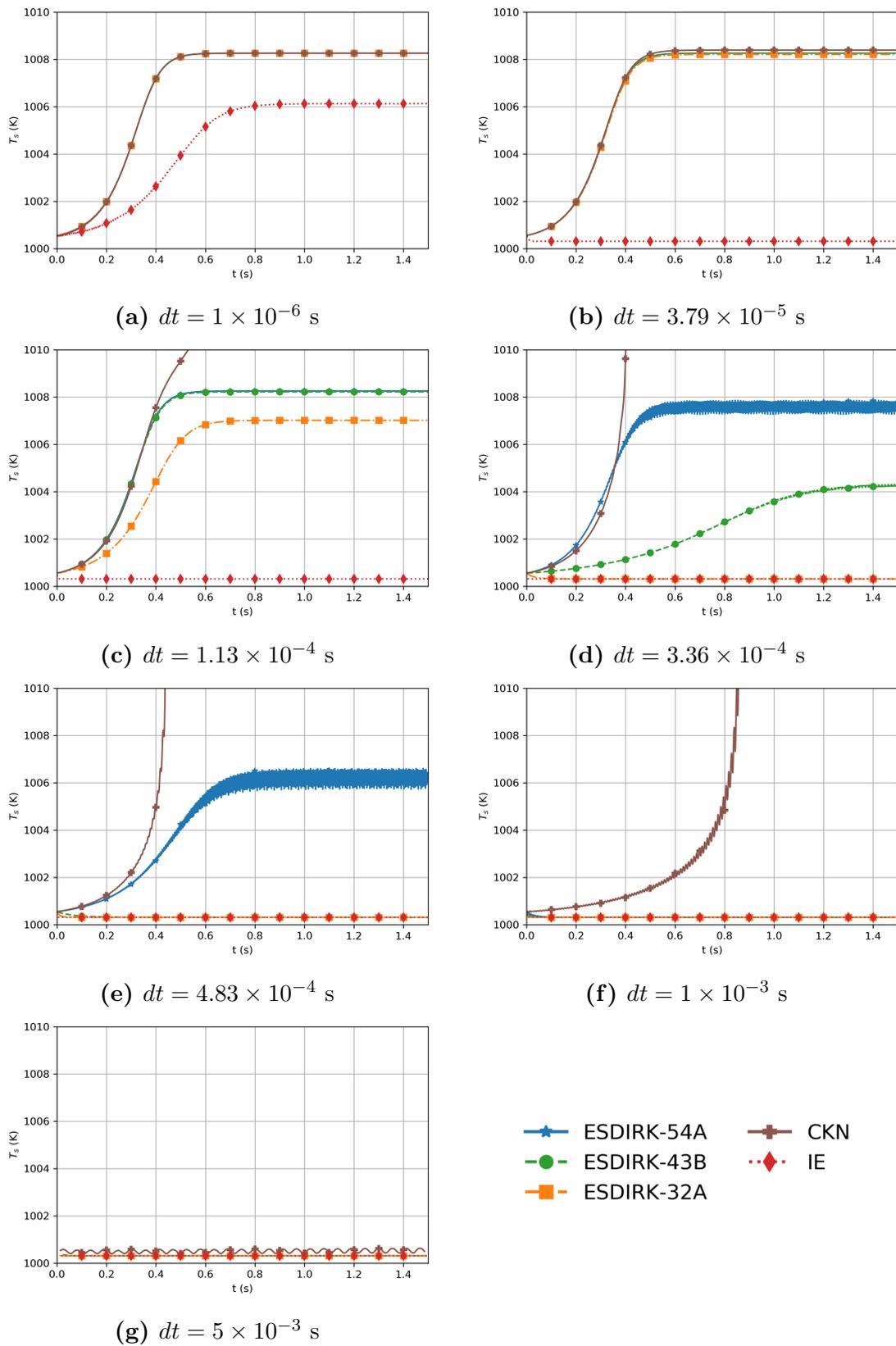
Finally, we see that the first-order IE is not able to correctly reproduce the limit cycle, even with the smallest time step of  $10^{-6}$  s. Using such a time step is already prohibitive, therefore lowering it further cannot be considered a viable solution to achieve an accurate result. Convergence results presented hereafter are obtained on the range  $dt \in [10^{-5}, 10^{-1}]$  s, where IE never produces an initial amplification and established limit cycle, therefore its results are omitted for the sake of readability.

#### Initial growth

Figure 6.9a shows the evolution of the fitted exponential amplification factor  $b$  as the time step is lowered. We see that all methods converge to the same value, however ESDIRK-54A and CKN are the first methods that manage to capture a growth (crossing the line  $b = 0$ ), and also the quickest to converge to the correct value.

Figure 6.9b shows the relative error of the amplification factor with respect to the reference solution. We see once again the same ranking in terms of ability to find the correct factor. At any time step  $\Delta t \leq 5 \times 10^{-4}$  s, ESDIRK-54A yields the best accuracy. Moreover we can observe that each method has an asymptotic convergence region where the order of convergence is close to the order of the method. In particular CKN, which initially performs well for moderate time steps, is quickly overtaken by the other methods that possess a higher convergence rate. However, we notice that both the order of convergence of the amplification factor for ESDIRK-43B and ESDIRK-54A is 4. This can be simply explained by the fact that  $b$  is determined via cubic interpolation of the curve of  $T_s(t)$  (see the methodology previously described), which introduces an error proportional to  $\Delta t^4$ .

For low time step values, the flattening of the convergence curves can be simply explained. The amplification factor is defined as a coefficient from an exponential fit, however this fit is only an approximation, as the nonlinear behaviour will let the unsteady evolution slightly deviate from the theoretical exponential initial growth. Also, the fit is based only on the successive maxima, not on the complete oscillating curve, which induces additional errors, e.g. imprecision in the abscissas



**Figure 6.8** Envelopes of the surface temperature histories computed for different time steps and integration methods

of the maxima. Therefore there is a point at which the precision achieved with an exponential approximation cannot be improved further.

### Limit cycle

The fundamental frequency of the established limit cycle is  $f \approx 452$  Hz. Figure 6.10a shows how the relative error on this frequency evolves with the time step, and Figure 6.10b shows the convergence of the amplitude of the fundamental frequency. The brown crossed curve for CKN is interrupted in the intermediate range of time step values, as the solution diverges, thus not allowing for an established limit cycle to be analysed. We can observe that ESDIRK-43B and ESDIRK-54A yield the most precise solutions at any given time step. In particular, ESDIRK-43B is able to capture a non-zero oscillation amplitude with larger time steps than required by the other methods. The frequency-finding process is not able to achieve unlimited accuracy in the determination of the fundamental frequency, hence the flattening of the convergence curves when the relative error reaches  $10^{-6}$ .

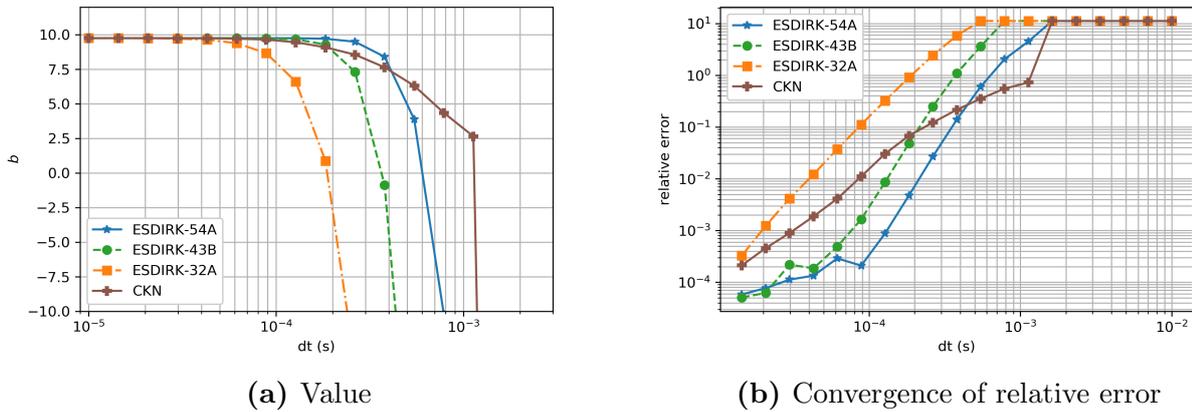


Figure 6.9 Fitted amplification factor

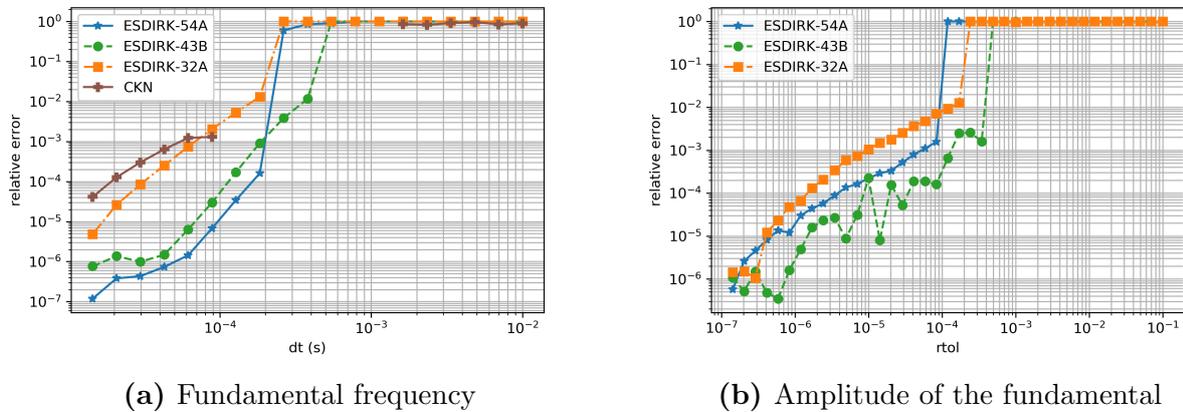
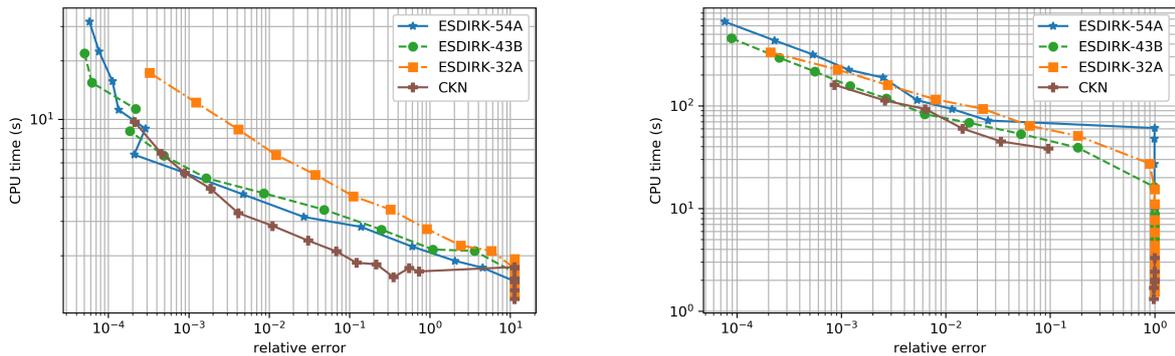


Figure 6.10 Convergence of the limit cycle properties



(a) Amplification factor (transient-only)

(b) Amplitude of the fundamental of the established limit cycle

**Figure 6.11** Work-precision diagrams for fixed time step simulations

### Computational cost

Based on the previous analysis, the high-order methods ESDIRK-54A and ESDIRK-43B seem particularly promising, as they require fewer time steps to achieve good results. However, due to the fact that these methods have more stages than the low-order methods, this does not mean that the actual computational times will be advantageous. Figure 6.11a shows the computational times versus the achieved relative error on the amplification factor. Note that these simulations were run only on the physical time interval  $t \in [0, 0.1]$  s, so that computational times are truly representative. We see that CKN is the fastest method for a relative error higher than  $5 \times 10^{-3}$ , however this roughly corresponds to the zone where the solution diverges in finite time. ESDIRK-32A is not a very good performer, whereas ESDIRK-43B and ESDIRK-54A are performing well and have similar error levels.

Figure 6.11b shows the computational time required for a given level of relative error on the fundamental amplitude in the established limit cycle. Computational times are those of simulations run on the physical time interval  $t \in [0, 1.5]$  s. ESDIRK-43B is only marginally better than the other methods, no clear winner is to be picked.

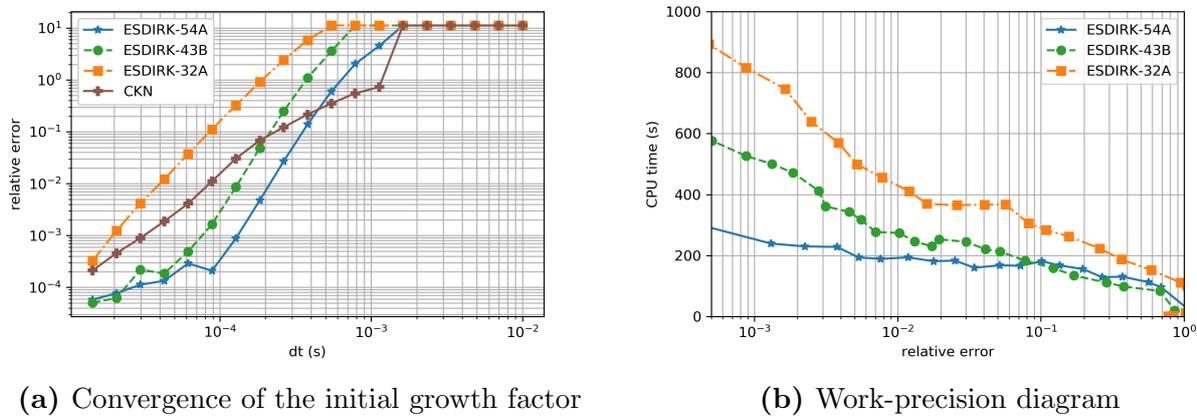
Overall, when using constant steps, the high-order methods ESDIRK-54A and ESDIRK-43B are almost identical and offer overall a very good performance. The Crank-Nicolson method is slightly misleading: its lack of stability when applied to DAEs leads to an easier destabilisation of the initial solution. However the method diverges quickly, unless the time step is very small, thus falsely leading to the conclusion of an unstable physical configuration.

### 6.3.4 Numerical experiment with time adaptation

The previous study with constant steps has shown that high-order methods are interesting for the simulation of a limit cycle. We now compare the embedded ESDIRK schemes with time adaptation enabled, to see if additional computational gains can be obtained. Following the methodology exposed in Section 5.1.5, the time step is controlled by the relative integration error tolerance  $rtol$ , which is varied between  $10^{-1}$  and  $10^{-7}$ . We first focus on the relative error achieved on the quantitative criteria used in the previous section. Finally, a comparison of the computational times and relative errors is presented, considering both fixed time step and adaptive simulations.

## Initial growth

Figure 6.12a shows the convergence of the fitted amplification factor  $b$  when  $rtol$  is decreased. We see that with fine tolerances, all three methods resolve the transient quite well. However, each method has a different onset of the convergence, for example ESDIRK-43B starts to properly resolve the transient amplification with  $rtol \approx 3.5 \times 10^{-4}$ , whereas as ESDIRK-54A needs at least  $rtol \approx 8 \times 10^{-5}$ . In Figure 6.12b, we plot the accuracy achieved on the amplification factor (which is not equal to  $rtol$ ) versus the computational time. We clearly see that, when the tolerance is sufficiently low, the computational cost decreases as the order of the method increases. For example, if we require a relative error of  $10^{-3}$ , ESDIRK-54A is twice as fast as ESDIRK-43B, and three times as fast as ESDIRK-32A. Only for high levels of error ( $> 10^{-1}$ ) is ESDIRK-43B slightly more efficient than ESDIRK-54A.



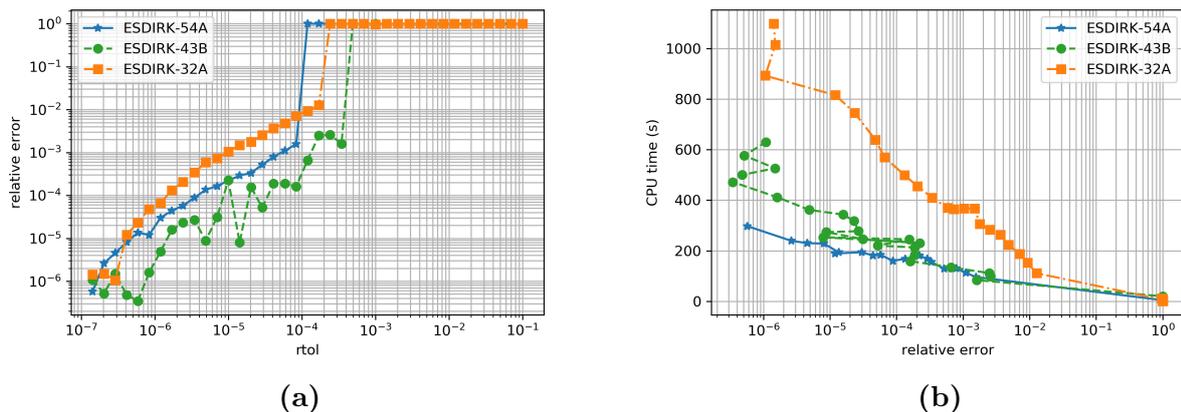
**Figure 6.12** Convergence and computational cost for the amplification factor

## Limit cycle

We now compare the computational cost of each method when considering the resolution on the time interval  $t \in [0, 1.5]$  s. We have observed that the frequency of the fundamental and its amplitude converge equally well, therefore we only focus on the amplitude. Figure 6.13a shows how the relative error on the fundamental amplitude evolves with  $rtol$ . ESDIRK-43B has, for a given  $rtol$ , the lowest error, however there is an unexplained oscillation of the relative error. Figure 6.13b is the corresponding work-precision diagram. ESDIRK-32A is the worst performer by far, whereas ESDIRK-54A is the most efficient method.

## Comparison with fixed time step results

We now wish to assess the performance gain achieved with time adaptation. To this end, we compare the computational times between fixed time step simulations and adaptive simulations for a given level of relative error. Figure 6.14a shows how the relative error on the amplification factor  $b$  during the initial growth evolves with computational time. The simulations were run for only 0.1 s of physical time, i.e. only for the initial growth. We observe that fixed time step simulations are always more efficient in this context. In particular, fixed time step implementations of the ESDIRK



**Figure 6.13** Limit cycle with adaptive time stepping: (a) convergence of the fundamental amplitude, (b) work-precision diagram

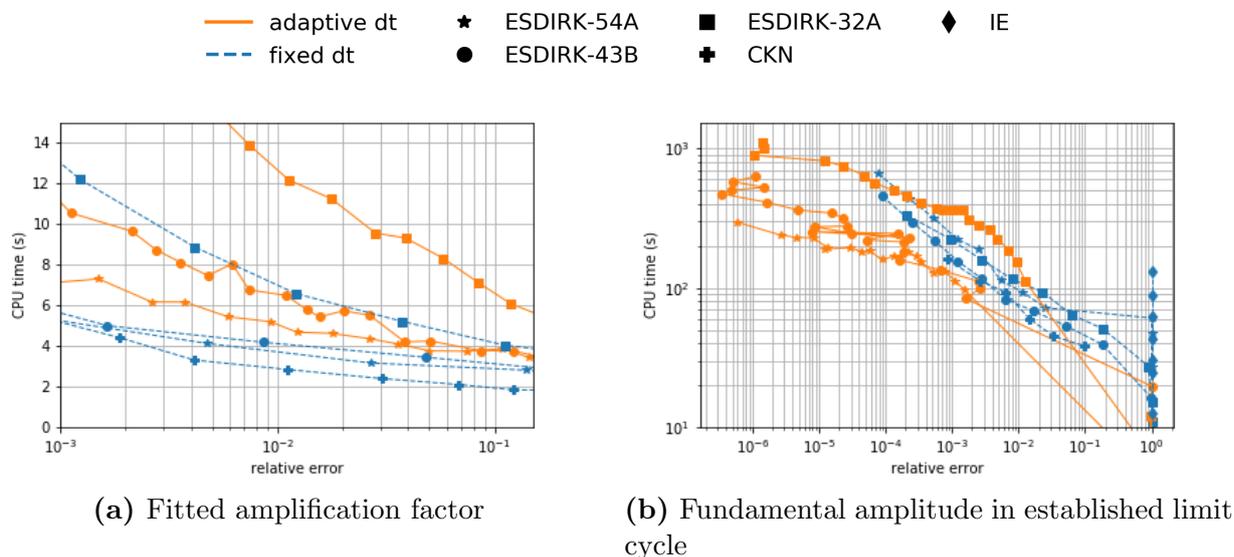
methods are faster than their adaptive counterparts. It has been assessed that this was due to an advantage in terms of the number of Jacobian evaluations for the Newton method.

Indeed, for the short time range simulated, the instability remains linear, i.e. the instability is linked with constant positive eigenvalues of the Jacobian of the system. Hence, this Jacobian does not need to be updated if the time step is constant. In practice, fixed time step solutions only required up to 3 evaluations of this matrix, while adaptive solutions required up to 100 evaluations due to the repeated changes in time step. It is possible to mitigate this issue by adding a “dead zone” for the time step evolution, i.e. refuse time step increases by less than a given percentage. In our testing, a 50 % dead zone yielded a 25% CPU time improvement on this test case, dividing by 3 the number of Jacobian evaluations. However, such a dead zone approach did not perform very well in the ignition test case from Section 6.2, hence we do not consider it further. Finally, the Jacobian update process could be improved. In the current version of VULC1D, every time the Jacobian (5.40) of the residuals needs to be updated, all its terms are computed anew. In the case of nearly linear dynamics, large computational gains could be expected from simply refactorising it upon each time step change, and only updating the stored Jacobian of  $f$  or  $g$  when the Newton does not converge.

The performance for the computation of the full limit cycle (initial growth and established cycle) is assessed in Figure 6.14b. The criterion is the relative error on the amplitude of the fundamental in the established limit cycle. One may think that the lightweight second-order Crank-Nicolson method could outperform the other methods for the established limit cycle, as this method is known to have good damping properties for oscillating systems, while only requiring one stage to be computed per time step. Indeed, the method is the fastest among the fixed time step ones, and the fastest overall for relative error levels around  $10^{-2}$ . For lower error levels however, the adaptive high-order methods ESDIRK-54A and ESDIRK-43B are the most efficient methods. ESDIRK-32A in adaptive mode is generally slower than in fixed time step mode, unless very low errors are sought. Although not shown here, adaptive simulations require more Jacobian evaluations due to the successive changes in time step, but they require many fewer steps and Newton iterations overall, which, for the complete simulation, far outweighs the drawback of evaluating the Jacobian more often. This is supported by the observation that the slopes of CPU time versus relative error are smaller for adaptive methods compared to fixed time step implementations. Any additional cost, e.g. Jacobian evaluation, is

overcome by the ability to take fewer steps. Still, adaptive methods are at a slight disadvantage in this test case, as the solution oscillates smoothly: the characteristic time scale of the system stays roughly constant throughout the simulation, therefore fixed time step simulations, with  $dt$  sufficiently low compared to this time scale, will be favoured by this consistency.

An interesting observation can be made: the adaptive methods always capture the correct limit cycle, unless  $rtol$  is too high, leading to a stabilisation of the solution. This is seen in Figure 6.14b, as all adaptive methods have a jump from important errors ( $\approx 1$ ) to much lower ones as  $rtol$  is lowered. On the opposite, fixed time step implementations do not have such a jump in error and are more likely to capture a non-accurate limit cycle (typically with an error higher than  $10^{-2}$ ) for an intermediate range of time step values. Overall, ESDIRK-54A and ESDIRK-43B seem to be the most reliable methods in this comparison.



**Figure 6.14** Comparison of the computational cost for a given level of relative error

Another practical consideration is that the time step corresponding to  $CFL = 1$  lies around  $10^{-6}$  s, which is approximately 100 times smaller than the time step necessary to obtain a very precise simulation with the fifth-order method ESDIRK-54A (see Figures 6.9b and 6.10b). The CFL-controlled time step is based on a stability analysis of the convection operator with an explicit time integration, which is not relevant for implicit integration and does not guarantee any level of error on the solution. As already discussed for the ignition transient in Section 6.2, use of a CFL-limitation would result in an important increase in computational time, without any valuable improvement on the solution accuracy. Finally, the constant time step simulations that yield accurate results more efficiently than with adaptive methods correspond to  $CFL \approx 100$ , which would usually not be expected to produce accurate unsteady results. One would rather safely choose a time step such that  $CFL = 1$ . This highlights one practical benefit of the time adaptation: precision is ensured based on a reliable mathematical criterion, and time step values can be used such that  $CFL \gg 1$ , while still ensuring a precise solution.

## 6.4 Application to unsteady combustion with detailed chemistry

We have now verified that the high-order adaptive methods perform well on relatively simple test problems. The next step is to test how they perform for the simulation of a propellant combustion with detailed chemistry. Additional stiffness is usually observed when a complex kinetic mechanism is used, thus it is useful to check the behaviour of the proposed numerical strategy in this context. The test case is the unsteady combustion of the AP monopropellant in a one-dimensional approach. The gas-phase kinetics is based on the AP-HTPB mechanism developed successively by Jeppson [35] and Tanner [165], initially for steady-state combustion. All reactions involving carbonated species were removed to account for the absence of HTPB, resulting in a pure AP combustion mechanism involving 25 species and 80 reactions. Gas-phase molecular transport is treated in a simplified manner by using mixture-averaged approximations. Species diffusion fluxes are expressed by a Fickian formula with effective diffusion coefficients and are corrected to ensure they sum up to zero. Thermodynamic and transport properties are computed beforehand by CHEMKIN routines [210] and stored as lookup tables.

The solid phase and the surface are handled as in [94]: the solid is assumed inert, and all decomposition and gasification reactions occur at the surface. There are two global surface reactions: a direct dissociative sublimation, and a quasi-equilibrium decomposition. The regression speed is defined by a pyrolysis law taken from [94], and the proportions of gaseous products generated by the surface reactions are adjusted to obtain the experimentally measured regression rates at 20 atm, following the approach of Meynet [119]. These modelling choices allow for the use of detailed combustion kinetics while remaining within the simplified framework of solid and surface representation from the unsteady model of Chapter 2.

The computational mesh has 49 cells for the solid phase, and 126 cells for the gas phase, distributed in a non-uniform manner so that steady-state gradients are well resolved. Starting from a steady-state solution at  $P = 20.265 \times 10^5$  Pa, we study the transient occurring after a pressure step to  $P = 20 \times 10^5$  Pa.

The goal in this Section is not to demonstrate a physically realistic model, but rather to test the proposed numerical strategy on a case that is representative of complex solid-propellant simulations.

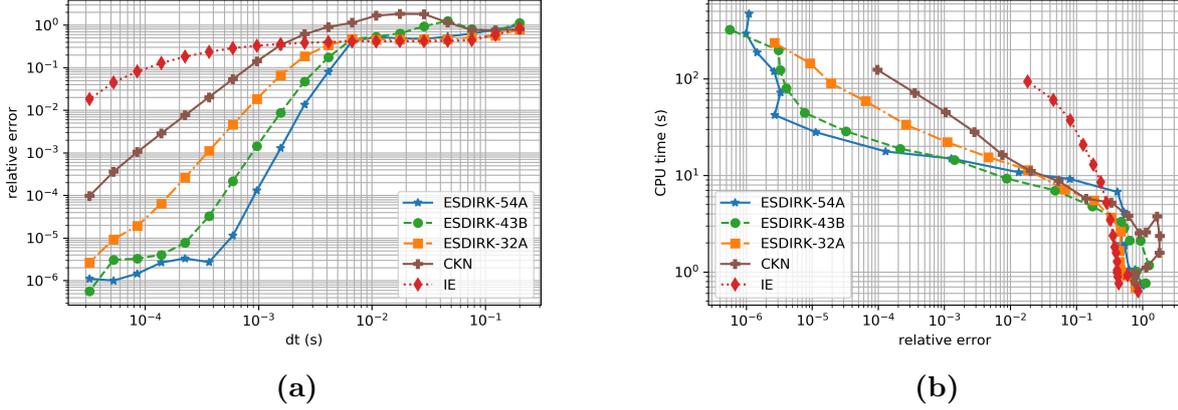
### 6.4.1 Order of convergence with fixed time steps

The orders of convergence of the methods from Table 6.1 have already been verified for the simple test case of Section 3.3.3 and are presented in 6.1.1. We now want to verify that the orders are not affected by the additional complexity and stiffness induced by detailed kinetics. We simulate the unsteady evolution for  $t \in [0, 0.2]$  s and perform multiple integrations with various time steps. To quantify the accuracy of the overall time integration, we define the following error:  $\epsilon_{T_s} = \frac{1}{t_f} \int_0^{t_f} |T_s - T_{s,ref}| dt$ , with *ref* designating the reference simulation and  $t_f$  the final physical time. The reference simulation is computed with ESDIRK-54A and  $\Delta t = 10^{-5}$  s. Cubic interpolation is used to compare both solutions on the same time grid.

Figure 6.15a shows the evolution of this error when the time step is varied. We see that each method attains its theoretical order of convergence. Similar results have been obtained for the other variables, both differential or algebraic, e.g. discrete mass flow rate field, cell temperatures. No order reduction is observed due to the stiffness of the chemical reactions with detailed kinetics. The

ESDIRK methods perform well in this more complex scenario.

Figure 6.15b shows for each method the computational time required to achieve a given integration error  $\epsilon_{T_s}$ . By analysing the different simulations, it was determined that the curve of  $T_s$  is visually converged when  $\epsilon_{T_s} \leq 10^{-2}$ . ESDIRK-43B and ESDIRK-54A are the most efficient methods on this error range, and the speed-up achieved by these high-order methods increases with the precision achieved. They are about 1.5 times faster than ESDIRK-32A and CKN for  $\epsilon_{T_s} = 10^{-2}$ , and approximately 5 times faster than CKN for  $\epsilon_{T_s} = 10^{-4}$ .



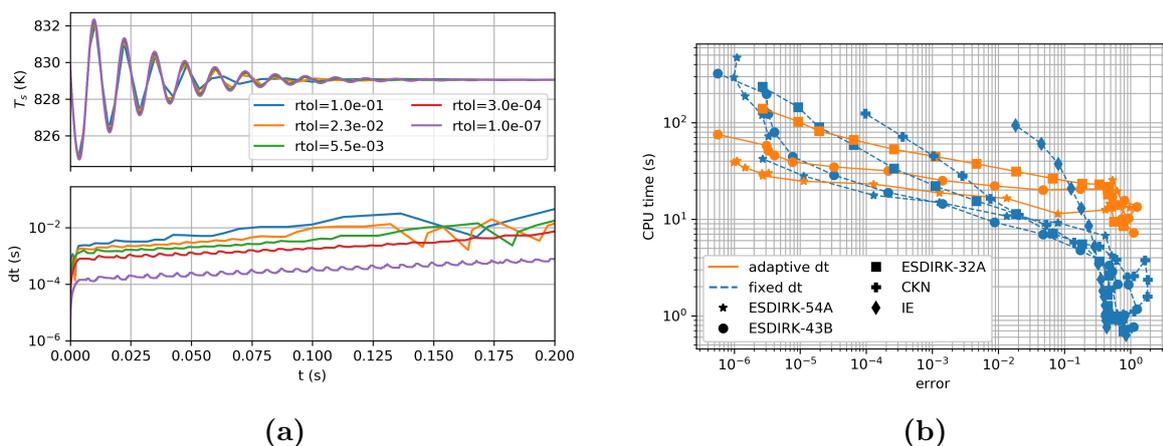
**Figure 6.15** Accuracy of the integration with fixed time steps: (a) Convergence of  $\epsilon_{T_s}$ , (b) Work-precision diagram for  $\epsilon_{T_s}$

## 6.4.2 Computational performance with time step adaptation

Now that we have verified that the convergence of the methods is not affected by the stiffness induced by complex kinetics, we use the ESDIRK methods with time step adaptation to see how they compare in terms of results. Different values of the relative integration tolerance  $rtol$  are used between  $10^{-1}$  and  $10^{-7}$ .

Figure 6.16a shows the complete transient for the surface temperature and the time step evolution for various values of  $rtol$ . We see that the change in time step is smooth, except for low tolerances when the time step becomes large, causing convergence issues. The temporal evolution of the surface temperature is well resolved even with relatively large values of  $rtol$ . Figure 6.16b shows the comparison of the computational time required to achieve a given level of error  $\epsilon_{T_s}$ , both with fixed time steps (blue lines) and adaptive time stepping (orange lines). Here, adaptive schemes do not seem to improve the performance globally. ESDIRK-54A is the best performing adaptive method, however it only becomes the fastest method overall for a very low level of error  $\epsilon_{T_s} \leq 10^{-5}$ . Its computational time is relatively close to the one of its fixed time step implementation. We observe that, for a given increase in accuracy, adaptive methods have a lower increase in computational time compared to their fixed time step counterparts.

This comparison is slightly unfair, because we had no *a priori* knowledge of the time step needed to properly resolve the transient. Also, the characteristic times of the dynamics does not vary much, as we can see from the small variations of the adapted time step in Figure 6.16a. Without dynamic time adaptation, the time step would have typically been limited so that the CFL number is reasonably low, e.g. 1 to 10. The time step corresponding to  $CFL = 1$  is around  $7 \times 10^{-8}$  s,



**Figure 6.16** Integration with adaptive time stepping: (a) time step evolution for ESDIRK-43B, (b) work-precision diagram for the time integral error  $\epsilon_{T_s}$

which is much lower than the time step required to achieve a good accuracy with most methods. A simulation has been carried out only on the first 0.03 s of the transient, with CKN and a time step set such that  $CFL = 10$ . The computational time was 660 s. An equivalently well resolved transient can be obtained with ESDIRK-54A and  $rtol = 10^{-5}$  in only 11 s. As we can see in Figure 6.16a, even a less stringent tolerance would also be sufficient. From an engineering point of view, this represents a 60 times speed-up, due to the fact that the time adaptation will automatically choose the relevant time step values. This adaptation ensures the problem is well resolved, while reaching CFL numbers that one would usually never trust to yield accurate unsteady results. Finally, time adaptation based on embedded methods automatically detects a slowdown of the dynamics as the solution stabilises and is able to increase the time step accordingly, whereas the CFL number stays roughly constant and cannot be an efficient time-step controlling criterion in that situation. This gain in engineering time is not quantifiable precisely, however it is definitely important.

## Partial conclusion

The VULC1D code has been verified in terms of temporal and spatial discretisations by comparison with theoretical results and with the semi-analytical tool from Chapter 3. Multiple test cases have then been used to demonstrate the capabilities of the VULC1D code.

The results show that stiffly accurate singly-diagonally implicit Runge-Kutta methods are highly efficient for the time integration of our 1D model, in particular the embedded ESDIRK methods presented in [68], handling the continuity equation with the approach from Section 5.2. Applications have been presented for ignition transients and limit cycle development with a simplified modelling, and appreciable computational gains have been observed. In particular, high-order methods can reliably capture nonlinear dynamics which are practically impossible to reproduce with traditional low-order methods. A test case of unsteady combustion with detailed stiff kinetics has also been set up, showing that numerical strategy from Chapter 5 is robust and performs well when the modelling is much more complex.

A strong point of the strategy is the dynamic adaptation of the time step based on objective mathematical criterion, which ensures proper resolution of the unsteady phenomena. From an engineering point of view, the single parameter that controls the time step is the relative integration

error tolerance  $rtol$ . In all our test cases, we have observed that  $rtol = 10^{-5}$  is sufficient to accurately resolve all unsteady phenomena. Using this value as standard tolerance liberates from the need of iterating over other practical criteria such as CFL limitation or maximum relative variation often used in the one-dimensional combustion community. We believe that a high-order adaptive method like ESDIRK-54A therefore allows for perceivable gains both in computational time, trustworthiness of the results, and the engineering time spent parametrising the time integration for a simulation. Furthermore, the proposed numerical strategy is easy to implement in an existing code if the latter already uses an implicit Euler, Crank-Nicolson or BDF scheme, as is often the case in the literature.

Eventually, the proposed time strategy is not dependent on the chosen spatial discretisation and other spatial schemes could be envisioned without changing the conclusion of our study. As mentioned in Chapter 5, the numerical strategy can be applied to any other one-dimensional combustion problem in the low-Mach limit involving homogeneous or spray combustion. When index-2 algebraic variables are involved, the order obtained on such variables is limited by the stage order of the method as indicated in [206, 218]. This situation occurs either for the strain rate eigenvalue for counterflow diffusion flames or in multidimensional Navier-Stokes equations, either incompressible or in the low-Mach limit [220, 240]; however the proposed strategy should be equally of interest in such cases.



# Part II

## Coupling



The aim of the present work is to simulate ignition of complete SRMs. This requires the reproduction of physical interactions between the chamber flow field and the solid propellant, in particular the conjugate heat transfer at the propellant surface and the parietal mass injection after ignition. A monolithic model, where both fluid and propellant models are strongly coupled, would be very complex to implement. Ideally, a better approach would be to couple two separate solvers: a 3D solid phase solver, that computes the unsteady heat conduction within the solid propellant and the surface state, with a 3D CFD tool, which simulates the propellant flame and the chamber flow field. However, there is a large difference in terms of spatial scales: 10-100  $\mu\text{m}$  for the propellant flame, a few meters for the whole combustion chamber. SRM simulations typically use mesh cells with a size close to 1 cm [50, 55, 56]. Solving the propellant flame accurately however requires mesh cells closer to 1  $\mu\text{m}$  or even less<sup>1</sup>, as typically used in detailed propellant solvers (1D to 3D) [33, 34, 36, 39, 137, 167]. Hence, it is computationally intractable to use a sufficiently well refined CFD mesh near the propellant surface across a complete motor in 3D, not to mention the cost of transporting all the required species and computing the associated reaction rates over the whole domain. In Chapter 1, a thorough review of the literature on that topic has shown that this issue is conveniently circumvented by using a 1D propellant combustion model at each CFD boundary point, which computes the unsteady evolution of the solid phase thermal profile and models the propellant flame in a simplified manner [50, 51, 54].

In Part I, we have presented the VULC1D code, which is specialised in the numerical resolution of a 1D model of solid propellant combustion. The well-established fluid flow solver CHARME of the multiphysics suite CEDRE from ONERA can be used to simulate the internal flow field of an SRM. Therefore, we propose to follow the approach of temporal splitting, i.e. letting each solver compute the unsteady evolution of their respective models, with regular exchange of coupling information (surface heat flux, mass injection...). This requires a coupling algorithm, which defines the accuracy and order of convergence of the overall solution in time, as well as its stability.

In our case, we shall insert the one-dimensional code VULC1D from Part I as a dynamic boundary condition in the CFD solver CHARME. Such a coupling has already been performed at ONERA [51], using earlier simplified ignition models, with first order accuracy in time overall. A number of adaptations are however required to couple the new 1D code. The novelty of our approach lies in the numerical resolution of the propellant flame within VULC1D, which enables the use of much more refined flame descriptions.

Thus, in Chapter 7, we shall discuss the key points that must be taken into account. First, the connection conditions are presented, and similarities with the field of conjugate heat transfer are discussed. Some issues arising from the one-dimensional flame model embedded in the boundary condition of CHARME are tackled. Another coupled approach is presented, where the propellant flame is solved within the CFD solver directly, leaving only the solid and surface resolution to VULC1D. This detailed approach makes it possible to conduct detailed simulations of small and simple combustion chambers. By comparison with the first approach, the effects of the one-dimensional modelling of the propellant flame and its compression into a boundary phenomenon for the CFD solver can be studied. The algorithm for the temporal coupling of VULC1D and CHARME is described, with first-order accuracy in time. As we will see in Chapter 11, devoted to applications, this approach yields an already efficient and robust proof of concept for the simulation of SRM ignition.

In Chapter 8, we shall investigate ways to increase the overall order of convergence in time of

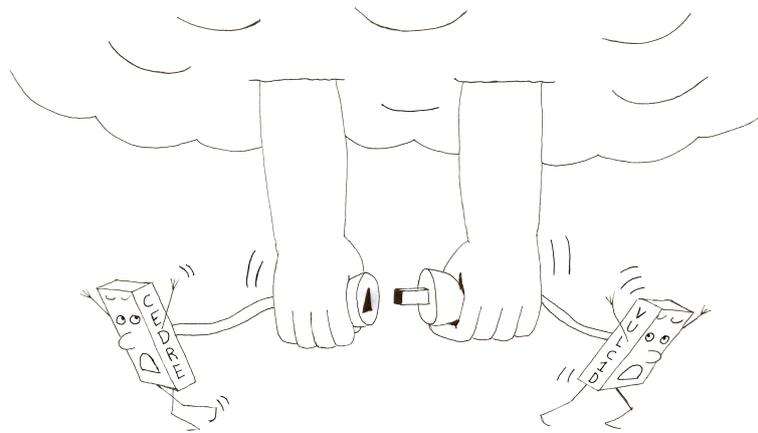
---

<sup>1</sup>Note that the mean free path in the chamber is typically 0.01  $\mu\text{m}$ , therefore the approximation of continuous medium is still valid for the propellant flame.

this coupling, to further improve the computational efficiency and solution accuracy. In particular, an algorithm that couples both fluid and propellant solvers with an adaptive time step and coupling order will be discussed, using polynomial extrapolations of the coupling variables and fluxes. This strategy is equally applicable to other problematics involving coupling of solvers via an interface, such as the simulation of conjugate heat transfer.

# Chapter 7

## Coupling the 1D model with a CFD code



**Figure 7.1** Artist's view of the coupling process

### Summary

For the simulation of SRM ignition, we propose to couple the one-dimensional code `VULC1D` with the CFD toolchain `CEDRE`. The physical interactions are described, and a first-order temporal coupling algorithm is constructed. To allow for detailed studies of the effects of modelling the propellant flame as a 1D boundary phenomenon for the CFD solver, an alternative approach is presented, where the propellant flame is directly solved within the CFD solver. Both implementations are verified by comparing their results with those of `VULC1D` for the one-dimensional laser-induced ignition of a solid propellant.

The physical interactions between the propellant and the chamber fluid models must be carefully handled so that each phase of an SRM ignition transient is adequately reproduced. In particular, the first phase is that of the inert heating of the propellant surface via impingement of hot igniter gases. A coupled simulation must therefore properly resolve the conjugate heat transfer (CHT) at the propellant surface before ignition, a problematic that already involves many difficulties and is still the subject of a large research effort [241–243]. The last phase, where the SRM behaves in a quasi-steady manner, occurs once the whole propellant surface has been ignited, and the mean chamber pressure has settled onto its steady-state value. At that point, the conjugate heat transfer to the solid propellant is purely controlled by the propellant flame, and the only influence of the

combustion chamber flow field on the propellant is via the wall pressure. In-between these two phases, the surface ignites in a progressive manner, a process which can be subdivided in two subphases: transition to ignition and flame spreading.

Flame spreading has been largely investigated in the ignition-specific literature [52, 54, 155] and it appears radiative preheating of the grain via the radiant emission of nearby ignited points is an important contributor to the rapidity of the flame spread. At ONERA in particular, such a phenomenon is modelled via the use of the radiative solvers REA or ASTRE, which are part of the CFD toolchain CEDRE [71], as used for instance in [155]. In the present manuscript, we discard this phenomenon to simplify our approach, however it can be easily taken into account thanks to the integration of the previous solvers in CEDRE, and the absorption of a radiative heat flux at the propellant surface or in-depth within the propellant can be easily included in VULC1D.

The other subphase, transition to ignition, is the process during which a given surface point switches from an inert behaviour to a reactive and transpiring surface one. During that phase, the flow rate of pyrolysis products increases and the propellant flame establishes itself above the surface point considered. As we have seen with the literature review from Chapter 1, most ignition models instantaneously switch from an inert behaviour to a quasi-steady burning rate injection, once the local surface temperature exceeds a user-defined ignition value. This means that flame establishment is assumed instantaneous. With the more dynamic ignition models from ONERA [27, 51, 52], there is however a phase during which the propellant flame develops. This is key to avoiding the use of a fixed ignition-temperature criterion, and for reproducing dynamic burning effects where the burn rate temporarily exceeds its steady-state value due to excessive pre-heating. During that transition phase, the source of heat for the solid propellant switches from heat conduction in the thermal boundary layer of the chamber flow field, to heat conduction from the newly appeared propellant flame. It is however not clear how these two contributions must be combined. Some studies at ONERA [51] have for instance considered a smooth transition based on the surface mass flow rate, however such an approach may prove problematic when finely resolved meshes are used, a point which will be discussed in greater details in Section 7.3.2.

Overall, compared to the more traditional CHT framework, the simulation of the heat transfer to the solid propellant adds multiple complexities:

- transition from an inert surface to a reactive surface,
- strong heat release from the propellant flame in a very thin zone above the propellant surface (typically a few hundred micrometers),
- parietal mass injection which depends in a strongly nonlinear manner on the surface temperature.

In addition to these difficulties, a few undiscussed issues arise from the one-dimensional flame model hidden within the boundary condition:

- the propellant flame (be it explicitly modelled with analytical transient models [51], conceptually hidden in semi-empirical burn rate laws [50, 54], or spatially discretised as in the present manuscript) has a size on the order of 100  $\mu\text{m}$  which means that it may well overlap with the first cells of the chamber CFD mesh. Thus, interactions between the internal flow field and the flame may not be properly captured. Additionally, the flame is assumed one-dimensional, thus potential lateral effects, such as thermal expansion or flame bending due to the longitudinal surrounding gas flow, are not accounted for.

- All the combustion models for large-scale ignition simulations use a quasi-steady propellant flame, or even a quasi-steady burning rate law (i.e. quasi-steady solid phase and propellant flame). As such, transient flame establishment may be inaccurate, in particular if the flame development time is not small compared to the characteristic time of ignition. Some numerical studies indicate that the flame establishment time is on the order of 10 ms for HMX [18] and AP-HTPB [39], however this depends on the pressure. Also propellant response to pressure oscillations will only be accurate at low frequencies for which the propellant (solid phase thermal profile and gas flame) may be considered quasi-steady. Based on the linear response (6.1) presented in Section 6.1, the quasi-steady solid phase assumption is only sensible for frequencies below 10 Hz for typical propellants approximately, which is too restrictive compared to the typical range of acoustic frequencies encountered in SRMs (10-1000 Hz).

Therefore, in this chapter, we first focus on the modelling aspects, and recall the compressible reactive fluid model used in the CFD code CHARME from the CEDRE suite. To investigate the previously discussed issues, it is valuable to have a detailed simulation framework where the propellant flame can be solved directly within the CFD solver, via proper resolution of the reactive multispecies Navier-Stokes equations, which is the “ideal” approach discussed in the introduction of the second part of this manuscript. We shall refer to it as the *detailed coupling*, or *detailed approach*. Although unsuited to 3D simulations of SRMs, it is expected that the detailed approach is tractable for the study of reasonable academic configurations in 1D or 2D. We present this approach, and discuss how the solid propellant and chamber flow models are coupled at the propellant surface. We then introduce the 1D flame coupled approach, using the one-dimensional gas phase model from Chapter 2 as a boundary model, discarding the flame from the CFD fluid model.

A one-dimensional comparison of these two approaches with VULC1D for laser-induced ignition verifies their proper implementation, and demonstrates that they yield nearly identical results. This indicates that the boundary model involving a 1D flame does not induce physical artefacts due to the additional modelling assumptions it involves. This is an important contribution, as that subject had previously not been addressed in the literature.

## 7.1 Physical models and interactions

In this section, we first present a detailed modelling of the propellant combustion, where the propellant flame and the combustion chamber flow field are captured within a single fluid model. This modelling will be used in the detailed coupling. We then introduce the one-dimensional gas-phase model from Chapter 2 as an additional sub-model to decrease the fluid model complexity.

### 7.1.1 Detailed model

The propellant combustion process, the chamber internal flow field and the conjugate heat transfer at the propellant surface can be represented by two models: one fluid model for the flow field (including gas phase reactions) and one for the evolution of the thermal profile inside the solid propellant. Both models are connected at the propellant surface via various balance equations, involving surface degradation of the solid. Let us present each model in detail, as well as the connection conditions.

## Fluid model

The combustion chamber gas flow is modelled by the multispecies compressible Navier-Stokes equations. We recall here the laminar conservation equations for mass, species, momentum ( $j$ -th direction), total energy  $e_t$ , written using Einstein notations, separating the Euler flow, diffusive, and source term contributions:

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho Y_{\mathfrak{k}} \\ \rho u_j \\ \rho e_t \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{\partial \rho u_i}{\partial x_i} \\ -\frac{\partial \rho u_i Y_{\mathfrak{k}}}{\partial x_i} \\ -\frac{\partial \rho u_i u_j}{\partial x_i} - \frac{\partial P}{\partial x_j} \\ -\frac{\partial \rho u_i e_t}{\partial x_i} - \frac{\partial P \delta_{ij} u_i}{\partial x_j} \end{pmatrix}}_{\text{div}(\vec{F}_{Euler})} + \underbrace{\begin{pmatrix} 0 \\ -\frac{\partial J_{\mathfrak{k},i}}{\partial x_i} \\ \frac{\partial \tau_{ij}}{\partial x_i} \\ -\frac{\partial}{\partial x_i} \left( q_{c,i} + \sum_{\mathfrak{k}=1}^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k},i} \right) + \frac{\partial \tau_{ij} u_i}{\partial x_j} \end{pmatrix}}_{\text{div}(\vec{F}_{diff})} + \begin{pmatrix} 0 \\ \omega_{\mathfrak{k}} \\ 0 \\ 0 \end{pmatrix} \quad (7.1)$$

The viscous stress tensor  $\tau_{ij}$  is computed as  $\tau_{ij} = 2\mu \mathcal{D}_{ij}$  where the deviator tensor is defined as  $\mathcal{D} = S - \frac{1}{3} \text{tr}(S) \mathbb{I}$  with  $S = \frac{1}{2} (\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^t)$ . The Euler fluxes  $\vec{F}_{Euler}$  gather the effects of convection and pressure (inviscid terms from the Euler equations), while the diffusive fluxes  $\vec{F}_{diff}$  gather the effects of heat, species and momentum diffusion. The conductive heat flux in the  $i$ -th direction is defined by Fourier's law as  $q_{c,i} = -\lambda \partial_{x_i} T$ . The mixture is composed of  $n_e$  species. The term  $J_{\mathfrak{k},i}$  is the component in the  $i$ -th direction of the diffusion flux vector  $\vec{J}_{\mathfrak{k}}$  for the  $\mathfrak{k}$ -th species. The volumetric production rate of the  $\mathfrak{k}$ -th species is  $\omega_{\mathfrak{k}}$ . The enthalpy  $h$  is the sum of the chemical and sensible enthalpies:  $h = \sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} h_{\mathfrak{k}}$ , where  $h_{\mathfrak{k}} = \Delta h_{\mathfrak{k}}^0 + \int_{T_{std}}^T c_{p,\mathfrak{k}}(a) da$ , with  $c_{p,\mathfrak{k}}$  the heat capacity of the  $\mathfrak{k}$ -th species, and  $\Delta h_{\mathfrak{k}}^0$  its formation enthalpy at  $T_{std}$  the standard temperature. The total energy is  $e_t = h + \frac{1}{2} u_i u_i - P/\rho$ . The mixture-averaged thermal conductivity  $\lambda$  and dynamic viscosity  $\mu$  are computed by simple arithmetic or harmonic means of the conductivity and viscosity of each species, weighted by their mass fractions, or in a mixed-manner as in Equation (2.18). The Kronecker symbol is  $\delta_{ij}$ . Soret and Dufour effects are neglected. The model does not account for gas-phase and surface radiation. Finally, the ideal gas law relates the pressure  $P$  to the various state variables in the gas phase:

$$\rho = P / \left( RT \sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}} \right) \quad (7.2)$$

with  $\mathcal{M}_{\mathfrak{k}}$  the molar mass of the  $\mathfrak{k}$ -th species, and  $R$  the universal gas constant.

Various formulations of the species diffusion flux may be employed. The Hirschfelder-Curtiss approximation [170, 171] can be used, where an equivalent diffusion coefficient  $D_{\mathfrak{k}}$  is defined as:

$$D_{\mathfrak{k}}^{eq} = \frac{1 - Y_{\mathfrak{k}}}{\sum_{j \neq \mathfrak{k}} X_j / D_{\mathfrak{k}j}} \quad (7.3)$$

The diffusive flux is then expressed as:

$$\vec{J}_{\mathfrak{k}} = -\rho D_{\mathfrak{k}}^{eq} \vec{\nabla} X_{\mathfrak{k}} \quad (7.4)$$

with  $X_{\mathfrak{k}} = \frac{\mathcal{M}}{\mathcal{M}_{\mathfrak{k}}} Y_{\mathfrak{k}}$  the molar fraction of the  $\mathfrak{k}$ -th species, where the mean molecular weight  $\mathcal{M}$  of the mixture is computed as  $1/\mathcal{M} = \sum_{\mathfrak{k}} Y_{\mathfrak{k}}/\mathcal{M}_{\mathfrak{k}}$ . This flux formulation is almost identical to that of

Equation (2.12) from the 1D model, which is slightly simplified to rely on the mass fraction gradient instead of the mole fraction gradient.

Equation (7.3) requires the specification of the binary mass diffusion matrix  $D = (D_{\mathfrak{k}j})$ . Since this is often difficult, a simpler implementation may be used, where a fixed Schmidt number  $\text{Sch}_{\mathfrak{k}} = \nu_{\mathfrak{k}}/D_{\mathfrak{k}}$  is prescribed for each species. Then, the diffusive flux for the  $\mathfrak{k}$ -th species reads:

$$\vec{J}_{\mathfrak{k}} = -\rho \frac{\nu_{\mathfrak{k}}}{\text{Sch}_{\mathfrak{k}}} \nabla \vec{Y}_{\mathfrak{k}} \quad (7.5)$$

In both cases, conservation of mass is ensured as in Section 2.1.2, by replacing the original diffusive fluxes  $\vec{J}_{\mathfrak{k}}$  with corrected fluxes  $\vec{J}_{\mathfrak{k}}^{\text{cor}}$  so that they sum up to zero:

$$\vec{J}_{\mathfrak{k}}^{\text{cor}} = \vec{J}_{\mathfrak{k}} - Y_{\mathfrak{k}} \sum_{j=1}^{n_e} \vec{J}_j \quad (7.6)$$

## Turbulence model

The flow in solid rocket motors is usually characterised by a large Reynolds number (typically  $10^6$ ). In this condition, it is known that the flow becomes turbulent, i.e. chaotic. Small variations in the boundary and initial conditions will lead to potentially large differences in the flow field. Each realisation of the flow is thus different. It is not tractable to compute each of them, thus a classical approach is to consider the Reynolds average of the flow. Each field  $q$  of the flow is decomposed as:

$$q(t, x) = \bar{q}(t, x) + q'(t, x) \quad (7.7)$$

where  $\bar{q}(t, x) = \mathbb{E}(q; t, x)$  denotes the statistical mean across all realisations of the flow (ensemble average), and  $q'$  is the fluctuation, which satisfies  $\bar{q}' = 0$ .

Each field  $q$  can be equivalently decomposed as:

$$q(t, x) = \tilde{q}(t, x) + q''(t, x) \quad (7.8)$$

where  $\tilde{q} = \overline{\rho q} / \bar{\rho}$  is the Favre-average and  $q''$  denotes the associated fluctuation, which satisfies  $\tilde{q}'' = 0$ . The Favre-average leads to simpler forms of the Reynolds-averaged Navier-Stokes equations (RANS) in the case of variable density flows [244].

Using these decompositions, it can be shown [245] that the momentum equation on  $\tilde{u}$  is:

$$\frac{\partial \bar{\rho} \tilde{u}_j}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{u}_j}{\partial x_i} = -\frac{\partial \bar{P}}{\partial x_j} + \frac{\partial (\bar{\tau}_{ij} - \overline{\bar{\rho} u_i'' u_j''})}{\partial x_i} \quad (7.9)$$

If we drop the  $\tilde{\cdot}$  and  $\bar{\cdot}$  notations, we obtain the non-averaged momentum equation from (7.1), but with the addition of the term  $-\overline{\bar{\rho} u_i'' u_j''}$  on the right-hand side, which can be interpreted as an additional stress, since it is grouped with the laminar stress  $\tau_{ij}$ . The matrix  $(-\overline{\bar{\rho} u_i'' u_j''})$ ,  $i, j = 1..3$ , is the autocorrelation matrix of  $u$ , also referred to as *Reynolds stresses*. Other terms involving  $u_i''$  also appear in the other conservation equations. All these unclosed terms need to be modelled.

Physically, the turbulent character of the flow means that fluctuations are present across a wide range of spatial scales. This is visible by the appearance of eddies, which are coherent local flow structures, whose sizes are distributed within the range  $[l_t, \eta_k]$ , with  $l_t$  the integral length scale (comparable to the size of the physical domain), and  $\eta_k$  the Kolmogorov scale. The fluctuations

of the flow velocity inside eddies can be associated with a fluctuating turbulent kinetic energy  $k$ , defined as:

$$k = \frac{1}{2} \overline{u'_i u'_i} \quad (7.10)$$

It has been observed that large eddies tend to transfer their kinetic energy to smaller structures, a process referred to as *energy cascade* [246]. At one point, the generated eddies are so small that viscosity becomes dominant and dissipates them. The size of the smallest eddies is therefore on the order of the length scale required to obtain a Reynolds number of 1, i.e. convective and viscous effects are comparable, which is the definition of the Kolmogorov scale  $\eta_k$ .

To improve the reproduction of the flow particularities (enhanced heat transfer, boundary layer development...), the effects of turbulence should be accounted for [245]. The ratio of the sizes of the largest eddies (integral length scale  $l_t$ ) and of the Kolmogorov scale is given by:

$$\frac{l_t}{\eta_k} = \text{Re}_t^{3/4} \quad (7.11)$$

where the Reynolds number  $\text{Re}_t = \rho U l_t / \mu$  is computed with  $l_t$  as reference scale, and  $U$  a reference velocity magnitude for the mean flow. For a typical flow field in a large SRM, the Reynolds number is close to  $10^6$ , hence small eddies are on the order of 30 000 times smaller than the largest ones. Thus, properly refining the CFD mesh to capture them is intractable.

To circumvent this issue, it is useful to interpret the energy cascade as an accelerator of viscous dissipation. In the laminar Navier-Stokes equations (7.1), the viscous effects are related to the dynamic viscosity  $\mu$  in the stress tensor  $\tau$ . This viscosity models momentum transfer at the molecular level, which can otherwise not be captured by a continuous flow model. Similarly, the dissipation of kinetic energy by small eddies, which are not resolved by the CFD mesh, can be modelled as the effect of a turbulent viscosity  $\mu_t$ , which augments the laminar viscosity  $\mu$ . This approach is commonly referred to as *turbulent-viscosity hypothesis* [245]. The determination of the local value of  $\mu_t$  is the main point addressed by the various RANS turbulence models presented in the literature. The kinematic viscosity  $\nu = \mu / \rho$  is the ratio of the square of a length scale (often referred to as the *mixing length*  $l_m$ ) and a time scale, which can be seen as two additional variables. Algebraic turbulence models define these scales based on the mean flow field or on educated user guesses, without introducing additional conservation equations into the Navier-Stokes system (7.1). One-equation turbulence models introduce an additional transport equation for the turbulent kinetic energy  $k$ , from which a velocity scale  $u^* \propto \sqrt{k}$  can be deduced, locally at each point in space. With the specification of the length scale  $l_m$ , the time scale can then be computed as  $l_m / u^*$ . Two-equation models avoid the specification of  $l_m$  by introducing an additional transport equation on the dissipation rate  $\epsilon$ , representing the rate at which  $k$  is dissipated thermally by turbulent viscosity in small eddies. This additional transport equation can also be on a function of  $\epsilon$  and  $k$ , for instance the turbulent frequency  $\omega = \epsilon / k$ . Thus, two transport equations are added to the Reynolds-average Navier-Stokes equations, from which  $k$  and  $\epsilon$  can be computed at every point in space. Then, the time scale is  $k / \epsilon$ , and the length scale is  $k^{2/3} / \epsilon$ .

The ubiquitous  $k - \epsilon$  model [247] has been used extensively for the study of turbulent flows. It usually performs well for free flows, however it fails at correctly reproducing near-wall flows and boundary layers. The  $k - \omega$  model, in particular its SST variant [248], has been shown to perform better for near-wall flows, while preserving the accuracy of the  $k - \epsilon$  model outside of the boundary layers. It is based on the use of a corrected formulation near the walls, while the free-flow region is modelled with the  $k - \epsilon$  approach. The transition between both formulations is smooth and is

a function of the distance to the nearest wall. The  $k - \omega$  SST model is known to behave well for separating flows, adverse pressure gradients and recirculation areas, compared to the  $k - \epsilon$  model and its other variants [248]. Such a turbulence modelling is used extensively for the simulation of SRM ignition [56, 58, 155] and is routinely employed at ONERA for simulations of the internal flow field of SRMs. In the rest of this subsection, we briefly present this model.

The  $k - \omega$  SST model is based on the following transport equations for  $k$  and  $\omega$ :

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_i k}{\partial x_i} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right] \quad (7.12)$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_i \omega}{\partial x_i} = \frac{\gamma}{\nu_t} \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\rho \sigma_\omega}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (7.13)$$

Both previous transport equations are used in conjunction with the Navier-Stokes equations (7.1). In the latter, each field  $q$  is replaced by its mean  $\bar{q}$ , the pressure  $P$  is replaced by  $\bar{P} + \frac{2}{3} \rho k$ , and the laminar dynamic viscosity  $\mu$  is replaced by the effective viscosity  $\mu + \mu_t$ .

Multiple constants appear in the above equations. A constant  $\Phi$  is computed as follows:

$$\Phi = F_1 \Phi_1 + (1 - F_1) \Phi_2 \quad (7.14)$$

The first set of values ( $\Phi_1$ ) is:

$$\sigma_{k1} = 0.5, \sigma_{\omega1} = 0.5, \beta_1 = 0.0750, \beta^* = 0.09, \gamma_1 = \frac{\beta_1}{\beta^*} - \sigma_{\omega1} \frac{\kappa^2}{\sqrt{\beta^*}}, \kappa = 0.41 \quad (7.15)$$

The second set of values ( $\Phi_2$ ) is the same as for the standard  $k - \epsilon$  model:

$$\sigma_{k2} = 1.0, \sigma_{\omega2} = 0.856, \beta_2 = 0.0828, \beta^* = 0.09, \gamma_2 = \frac{\beta_2}{\beta^*} - \sigma_{\omega2} \frac{\kappa^2}{\sqrt{\beta^*}}, \kappa = 0.41 \quad (7.16)$$

The turbulent viscosity is computed as:

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, \Omega F_2)} \quad (7.17)$$

with  $\Omega$  the absolute value of the vorticity, and  $a_1 = 0.31$ .

The term  $F_1$  and  $F_2$  are blending functions, which allow to dynamically switch between the  $k - \epsilon$  formulation for free flows and the corrected  $k - \omega$  model for near-wall flows. They are computed via complex formulae [248], which produce a behaviour in the form  $F = \tanh(1/y^p)$ , with  $p$  a positive integer and  $y$  the distance to the closest wall, so that  $F_1$  and  $F_2$  tend to 0 in the free flow.

## Solid propellant model

This model describes the evolution of the thermal profile within the solid propellant. Due to the insulating nature of the propellant, the thickness of the thermal profile is typically on the order of 100  $\mu\text{m}$  and is very small compared to other characteristic scales of the flow and motor. Consequently, we have already seen in Chapter 1 that most ignition models from the literature assume that heat diffusion only occurs in the direction normal to the surface, neglecting lateral heat transfer within the solid. We follow the same approach, and use the assumptions and models from Chapter 2.

Following the one-dimensional heat transfer assumption, we only describe the thermal profile by the temperature field  $T((x_i, i = 1..3), \eta)$ , with  $(x_i, i = 1..3)$  the position of the surface point considered in the frame associated with the combustion chamber, and  $\eta$  the coordinate along the normal to the surface. The propellant surface corresponds to  $\eta = 0$  and the solid phase is at  $\eta < 0$ . The thermal profile evolution is described by Equation (2.7), which we recall in the following with the alternative notation  $\eta$  instead of  $x$  for the propellant depth, to avoid confusion with the space variables from the fluid model. The temperature field  $T$  is subject to:

$$\rho_c c_c \partial_t T + \rho_c c_c r \partial_\eta T - \partial_\eta (\lambda_c \partial_\eta T) = 0 \quad (7.18)$$

with  $\rho_c$  the propellant density,  $c_c$  its heat capacity,  $\lambda_c$  the thermal conductivity, and  $r$  the absolute surface regression speed, deduced from Equation (2.25). Deep in the solid, the propellant is at temperature  $T_0$ , as per Equation (2.8).

### Surface connection conditions

The fluid and solid models are connected at the surface of the solid propellant, where we assume that all the pyrolysis-related phenomena are concentrated. Let us first consider the surface balance along the normal vector. We use the subscript  $s$  to identify the values that are taken at the surface for the solid propellant model, i.e. at  $\eta = 0$ . The connection conditions for the gas and solid phases at the interface are the same as Equations (2.21) to (2.24) used for the purely one-dimensional propellant combustion model. We recall them here in a more generic form, accounting for the multidimensional character of the chamber fluid model:

$$\left\{ \begin{array}{l} \rho_s u_s = m - \rho_s r = \rho_w \vec{u}_w \cdot \vec{n} \end{array} \right. \quad (7.19)$$

$$\left\{ \begin{array}{l} T(0^-) = T(0^+) = T_s \end{array} \right. \quad (7.20)$$

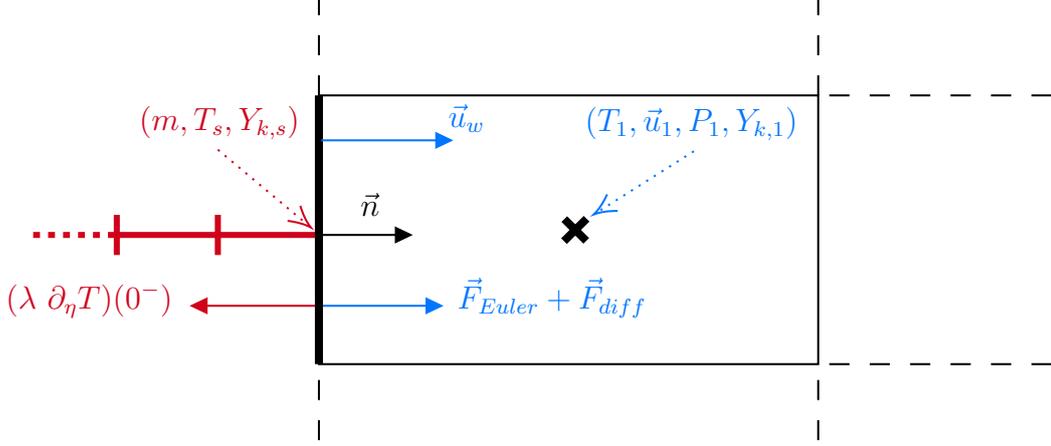
$$\left\{ \begin{array}{l} (mh - \lambda_c \partial_\eta T)_{0^-} = \vec{n} \cdot \left( mh \vec{n} + \vec{q}_c + \sum_1^{n_e} h_{\mathfrak{k}} \vec{J}_{\mathfrak{k}} \right)_{0^+} \end{array} \right. \quad (7.21)$$

$$\left\{ \begin{array}{l} (mY_{in,j,\mathfrak{k}})_{0^-} = \vec{n} \cdot \left( mY_{\mathfrak{k}} \vec{n} + \vec{J}_{\mathfrak{k}} \right)_{0^+} \quad \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket \end{array} \right. \quad (7.22)$$

with  $\vec{n}$  the normal unit vector oriented towards the  $\eta > 0$  (inside of the CFD domain),  $m$  the mass flow rate given by the pyrolysis law (2.25),  $\vec{u}_w$  the surface fluid velocity in the fluid domain, and  $\vec{q}_c = -\lambda \vec{\nabla} T$  the heat flux vector. The connection conditions and the geometrical configuration are sketched in Figure 7.2, where the 1D model appears in red, and the CFD fluxes and variables in blue. The cell center of the first CFD mesh cell above the surface (subscript 1) is depicted by the black cross.

In the simulations presented further in this manuscript, we only consider finely resolved meshes such that the turbulent viscosity  $\mu_t$  is correctly driven to 0 at the surface. Indeed, the mesh spacing required in practice to properly resolve the propellant flame is such that the near-surface turbulent flow field is very well resolved. This enables us to only consider the laminar expression of the fluxes at the surface. Otherwise, unresolved effects of turbulence should be accounted for by wall laws [245], which we do not consider in the present manuscript.

Finally, we need to specify boundary conditions for the fluid model, which is multidimensional. Hence additional conditions are required, in particular for the momentum equation. We use the subscript  $w$  to denote the parietal (wall) values used in the fluid model which, in the case of this detailed modelling, will be identical to the values at the propellant surface model (subscript  $s$ ). We assume the gaseous species that leave the surface are injected in the direction perpendicular to the



**Figure 7.2** Sketch of the connection conditions between the 1D model and the CFD mesh

propellant boundary face. We impose a no-slip condition at the surface, i.e. the fluid flow velocity at the surface  $\vec{u}_w$  must be perpendicular to the propellant surface:

$$\vec{u}_w = \|\vec{u}_w\| \vec{n} \quad (7.23)$$

where the norm of gas flow velocity is the value of the fluid velocity observed in the 1D model at the surface:

$$\|\vec{u}_w\| = \frac{\rho_s}{\rho_w} u_s \quad (7.24)$$

We remind the reader that, in this detailed model, the subscript  $s$  and  $w$  are equivalent, thus  $\|\vec{u}_w\| = u_s$ . Finally, Equation (7.19) leads to:

$$\|\vec{u}_w\| = \frac{m - \rho_w r}{\rho_w} = r \left( \frac{\rho_c}{\rho_w} - 1 \right) \quad (7.25)$$

Typically,  $r \approx 10^{-2}$  m/s and  $\rho_c/\rho_w \approx 50$ , thus  $\|\vec{u}_w\|$  is on the order of 0.5 m/s at the propellant surface.

The Euler boundary fluxes for the fluid model can then be expressed as:

$$\left( \vec{F}_{Euler} \right)_w = \begin{pmatrix} m \\ m Y_{\mathfrak{k},w} \\ m \vec{u}_w + P \vec{n} \\ m e_{t_w} + P \|\vec{u}_w\| \end{pmatrix} \quad (7.26)$$

We remind the reader that, in this detailed model, the subscripts  $w$  (values of fluid-model fields at the propellant boundary of the CFD domain) and  $s$  (values at the propellant surface in the propellant surface) are equivalent. However, since that will not be the case any more when the one-dimensional flame model is used, we prefer to make a clear distinction, so as to avoid confusion later on.

The surface viscous fluxes are expressed following their formulation in the complete Navier-Stokes equations:

$$\left( \vec{F}_{diff} \right)_w = \begin{pmatrix} 0 \\ \vec{J}_{\mathfrak{k}}^{cor} \cdot \vec{n} \\ 2\mu \vec{D}_i \cdot \vec{n} \\ \lambda \frac{\partial T}{\partial n} + \vec{u}_w \cdot \left( 2\mu (\vec{D}_i \cdot \vec{n}) \vec{e}_i \right) + \sum_{\mathfrak{k}} h_{\mathfrak{k}} \vec{J}_{\mathfrak{k}}^{cor} \cdot \vec{n} \end{pmatrix}_w \quad (7.27)$$

where  $\vec{\mathcal{D}}_i$  denotes the  $i$ -th column vector of the deviator tensor  $\mathcal{D}$ .

Finally, the parietal values of the turbulent variables  $k$  and  $\omega$  are user-specified constants, imposed as Dirichlet conditions for the CFD fluid model. Typically, we assume that no turbulence is introduced by the parietal mass injection, therefore we set the surface turbulent kinetic energy to 0.

### 7.1.2 One-dimensional propellant flame modelling

Following our previous discussions, the propellant flame has a small enough height such that the flame can be partially decoupled from the chamber flow field. To discard the flame-related kinetics from the fluid model, and relax the mesh refinement required near the surface of the propellant, it seems sensible to introduce an additional submodel that solves the propellant flame with a one-dimensional approach, acting as surface condition for the chamber fluid model. The low-Mach gas-phase model from Chapter 2 is therefore particularly appropriate. For the sake of clarity, we recall its main equations here, with the new notation  $\eta$  for the space coordinate along the height of the one-dimensional flame, as for the solid phase. The flame at  $\eta > 0$  is governed by the following partial differential equations:

$$\begin{cases} \partial_t \rho + \partial_\eta(\rho(u+r)) = 0 & (7.28) \end{cases}$$

$$\begin{cases} \partial_t \rho Y_{\mathfrak{k}} + \partial_\eta(\rho(u+r)Y_{\mathfrak{k}}) = -\partial_\eta J_{\mathfrak{k}} + \omega_{\mathfrak{k}} & \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket & (7.29) \end{cases}$$

$$\begin{cases} \partial_t \rho h + \partial_\eta(\rho(u+r)h) = -\partial_t P - \partial_\eta(q_c + \Sigma_1^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k}}) & (7.30) \end{cases}$$

This flame model is inserted as an additional modelling layer between the fluid and surface models. The surface connection conditions must now be applied between the solid model from Section 7.1.1 and the 1D flame model. The field variables of the 1D flame model replace the ones of the chamber fluid model in the right-hand sides of Eqs. (7.19) to (7.22).

New connection conditions must be provided to link the 1D flame and chamber fluid models. In the 1D model, the thermodynamic pressure is uniform due to the low-Mach assumption. To have a coherent interaction with the chamber flow, its value must be set to the parietal pressure  $P_w$  prescribed by the latter. In return, the 1D flame model transmits fluxes of mass, species and energy. From the point of view of the chamber fluid model, the propellant surface is still considered as an adherence surface, however the parietal values of the field variables are set to the values of the same fields at the exit of the 1D gas phase (subscript  $f$ ), instead of those at the propellant surface (subscript  $s$ ).

In practice, to ensure the propellant flame is entirely captured within the 1D flame model, the 1D domain extends much further than the typical flame height, so that chemical equilibrium is reached. This allows for flame-related kinetics to be discarded from the CFD fluid model. Consequently, the exit boundary conditions for the 1D flame are simple Neumann conditions:

$$\partial_\eta T(+\infty) = 0, \quad \partial_\eta Y_{\mathfrak{k}}(+\infty) = 0 \quad \forall \mathfrak{k} \in \llbracket 1, n_e \rrbracket \quad (7.31)$$

Therefore, no diffusive fluxes leave the 1D flame model, only convective fluxes. To ensure the coupling is physically coherent, surface species diffusion fluxes are set to 0 for the fluid model as well.

Following the introduction of the 1D flame model, the conjugate heat transfer between the fluid and solid models cannot be captured by the 1D flame model. Indeed, no heat can diffuse from the fluid model to the 1D flame model via the exit conditions of the latter. If the previous Neumann

conditions were modified to allow for non-zero diffusive fluxes at the boundary of the 1D flame domain, the conjugate heat transfer would then be dependent on the size of the 1D flame mesh, which is not acceptable. This is closely related to the issues discussed further in Section 7.3.

As a consequence, the conjugate heat transfer must be handled by reintroducing a direct connection between the fluid and solid models, bypassing the 1D flame model. This is done by inserting the wall heat flux from the fluid model into the surface coupling condition (7.21). This wall heat flux is computed as:

$$\Phi_w = (\lambda \vec{\nabla} T)_w \cdot \vec{n} \quad (7.32)$$

We stress that the temperature field used in this last equation is the one of the fluid model, not the 1D flame. It is then added to the surface enthalpy balance (7.21), forming the following modified equation:

$$(mh - \lambda_c \partial_\eta T)_{0-} = (mh - \lambda \partial_\eta T + \Sigma_1^{ne} h_t J_t)_{0+} - \Phi_w \quad (7.33)$$

All the variables in this equation, except  $\Phi_w$ , come from the solid and 1D flame models. Specific issues related to this modified surface enthalpy balance are discussed in Section 7.3.2. Note that surface friction power also increases the wall heat flux, however its contribution in the low-Mach near-surface flow is usually 3 to 5 orders of magnitude lower than that of the conductive heat flux, hence we do not include it in Equation (7.32) for simplicity.

Finally, the surface viscous fluxes for the fluid model are expressed by assuming adherence at the wall, where the flow is locally perpendicular to the propellant surface. The new surface velocity vector for the chamber fluid model is:

$$\vec{u}_w = \frac{\rho_f u_f}{\rho_w} \vec{n} \quad (7.34)$$

In practice, part of the intent of having such a model is to reduce the CFD mesh refinement for the fluid model near the wall. This is possible thanks to the flame not appearing within the fluid domain any more. In the RANS framework, further gains could be obtained by reducing the mesh refinement below the one classically required for an accurate computation of the conjugate heat-transfer. This would however necessitate the use of wall laws to account for the evolution of the thermal profile at scales lower than the height of the first cell above the surface. This introduces additional complexity to the model, therefore we do not consider that issue further in this work.

## 7.2 Coupled framework for SRM ignition simulations

To simulate the previous models and their interactions, we choose to avoid a monolithic approach where all the models are solved within one single solver. We rather capitalize on the codes already developed at ONERA, only requiring the specific surface coupling conditions to be implemented. In this section, we first present the CFD solver used, and describe the coupling procedure.

### 7.2.1 Description of the CFD solver

The fluid model for the combustion chamber flow field is implemented in the CFD solver CHARME, which is fully integrated in the semi-industrial multiphysics simulation toolchain CEDRE from ONERA [71]. CHARME uses the cell-centered finite volume technique on general unstructured meshes to semi-discretise in space the set of conservation equations (7.1). A MUSCL-type reconstruction method of order 2 to 4 can be used to increase the order of accuracy in space for the Euler flow contributions (convection and pressure terms). This method is used to compute the values of the

field variables on each side of every mesh face, and various limiters can be applied to ensure the total-variation-diminishing (TVD) character of the scheme. To compute the Euler fluxes at each face, the associated Riemann problems are solved with a variety of methods, e.g. the approximate solver HLLC [249], or Roe flux-vector splitting [250]. The diffusive fluxes are evaluated using a second-order centered scheme. In this manuscript, all simulation are performed with a second-order MUSCL scheme with multislope reconstruction [251]. A wide variety of boundary conditions are available, e.g. symmetry, heat transfer, non-reflection, porous walls.

For the temporal discretisation, several linearised implicit Runge-Kutta methods can be applied, of order one or two. The linear system obtained at each Runge-Kutta stage step is solved with the Generalised Minimal Residual method (GMRES) [252]. Explicit Runge-Kutta methods from order 1 to 4 are also available, however they will not be used for the present work, due to the excessive stability restrictions when dealing with fine meshes. Parallelisation is enabled via the use of MPI directives.

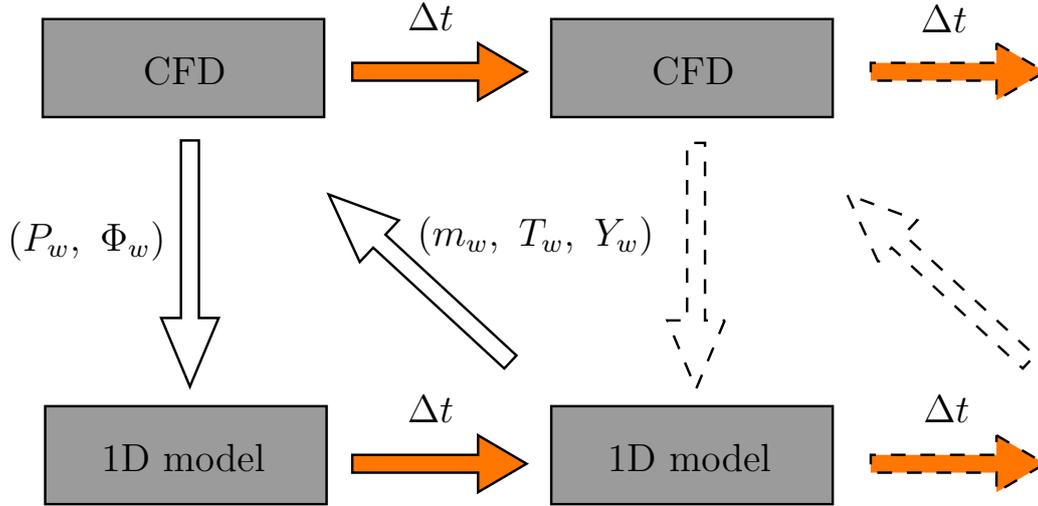
Thanks its integration within CEDRE, CHARME can be coupled with other codes, such as radiation or polydisperse spray solvers to model a wide variety of physical phenomena. CEDRE is routinely used by ONERA and collaborators for a wide variety of studies [253], e.g. subsonic to hypersonic aerodynamics, combustion chambers, multiphase flows.

## 7.2.2 Coupling methodology

The CFD solver CHARME and the solid propellant solver VULC1D can be coupled to compute the ignition of a combustion chamber. The underlying idea is the same as for the various large-scale ignition simulations presented since the 1990s [50, 56]. The combustion chamber is meshed in 2D or 3D and its internal flow field is simulated with the CFD solver. At each boundary face corresponding to the propellant surface, an instance of the one-dimensional propellant solver is used to compute the evolution of the thermal profile within the propellant and the combustion of the latter. At each time step of the coupled computation, the coupling variables and fluxes are exchanged between both solvers.

Following the various modelling levels presented in the Section 7.1, two approaches are designed. The first approach, the *detailed coupling*, uses the most detailed modelling, with the propellant flame being handled in the fluid model. VULC1D is used to compute the evolution of the thermal profile inside the propellant and of the surface variables, but its 1D gas phase is discarded. The 1D model provides the surface variables (temperature, mass flow rate, composition) and surface fluxes which are used as boundary conditions for the fluid model.

In the second approach, the *1D flame coupled approach*, the propellant flame is encompassed within the model from Section 7.1.2. The fluid model transmits the values of the parietal pressure and heat flux. However, for the fluid model boundary condition, the parietal variables (subscript  $w$ ) are set to their values at the end of the 1D gas domain (subscript  $f$ ) instead of at the propellant surface (subscript  $s$ ). This approach can be viewed as a generalisation of the much simpler ignition models from the literature already mentioned in the introduction. These use a quasi-steady burn rate law to avoid a more complex representation of the propellant flame, however the modelling choice is essentially the same: the flame is represented within the boundary model, not within the CFD domain. Thus the CFD domain does not need to be meshed very finely near the surface, and the CFD solver can work only with inert combustion products, sparing the computational expense of evaluating chemical source terms over the whole domain. Compared to other ignition models from the literature, an important contribution of our approach is the ability to use a much more



**Figure 7.3** Coupling algorithm between VULC1D and CHARME

refined description of the propellant flame if required, for instance by including detailed transport and kinetics. Also, our approach makes it possible to use global mechanisms with no simplifying assumptions on the activation energies of the various reactions, which should enable an accurate representation of a broader class of propellants.

### 7.2.3 Coupling in time

To produce a time-accurate simulation of the overall combustion chamber ignition, the previous flux exchange must be performed periodically. This is done at each so-called coupling time step, as depicted in Figure 7.3. At each time step, the CFD solver CHARME transmits the parietal pressure  $P_w$ , the heat flux  $\Phi_w$  (and potentially a radiative heat flux) to the propellant solver VULC1D. This solver then proceeds forward in time for one coupling time step, and gives back to CHARME a flow rate of mass, species, and enthalpy, which are injected through the previously described boundary conditions in CHARME. The latter can then move forward one step, and the whole process starts over again for the next coupling time step.

To improve the stability and precision of the computation, while also easing the simulation set-up, the coupling time step is dynamically selected such that, between two time steps, the relative variations of the field variables ( $\rho, u, T$ ) are below a prescribed threshold (e.g. 2%) in all CFD cells. Thus the near-surface cells typically dictate the overall time step. Technically, this time step control strategy has been implemented by using the local-time stepping capability of CHARME, which is used to improve the convergence towards steady-state solutions, by letting the CFD cells evolve with different time steps. However here, to maintain a coherent resolution of the unsteady dynamics, each cell is given a time step equal to the minimum of the local time steps computed over the whole domain. This increases the robustness and precision of a simulation. Indeed, potential numerical instabilities cause a time step reduction which leads to a rapid damping of these instabilities.

The coupling algorithm is first-order in time, and corresponds to a single Gauss-Seidel iteration of a waveform relaxation approach [254, 255]. Therefore, we only use a first-order accurate time scheme (backward Euler) in CHARME and VULC1D for the coupled simulations, and each solver only performs one time step per coupling step. The aim of Chapter 8 will be to find an easily implementable way to generalise this approach to higher-orders, tackling issues such as dynamic

time step and order adaptation, stability and convergence.

The coupling fluxes are exchanged explicitly, i.e. they are fixed during the integration of each subsystem. The stability of such a coupling has been studied in the framework of conjugate heat transfer, considering thermal diffusion as the only physical mechanism [241, 256], and it has been shown that instabilities may arise if the time step is too large. In Chapter 11, we will see that instabilities may indeed occur when the CFD gas-phase mesh is too highly refined near the surface.

## 7.3 Modelling issues

To ensure the flame is entirely captured, the 1D gas mesh from the solid propellant model is extended far enough so that equilibrium is reached at its exit, with a typical length of  $L = 1$  mm. Thus, all reactants are consumed and only non-reactive gases are injected into the CFD domain, sparing the computational expense of computing kinetics-related source terms in the CFD solver. This approach is coherent with the other ignition models from the literature [7, 51] where the flame, be it modelled by an analytical formula or hidden within a semi-empirical burn rate formula, is entirely contained within the solid propellant model.

However, the large extension of the 1D gas flame mesh means that there is a strong overlap between the 1D gas domain and the CFD domain. Figure 7.4 represents the spatial overlap, as well as the locations where the exchanged fluxes are applied. The cell centers of the 1D solid and gas meshes are represented by the red squares. The cell faces of the CFD mesh are represented by the thin vertical black dashed lines, and the cell centers are represented by the blue crosses. The temperature at the center of the first CFD cell is  $T_1$ .

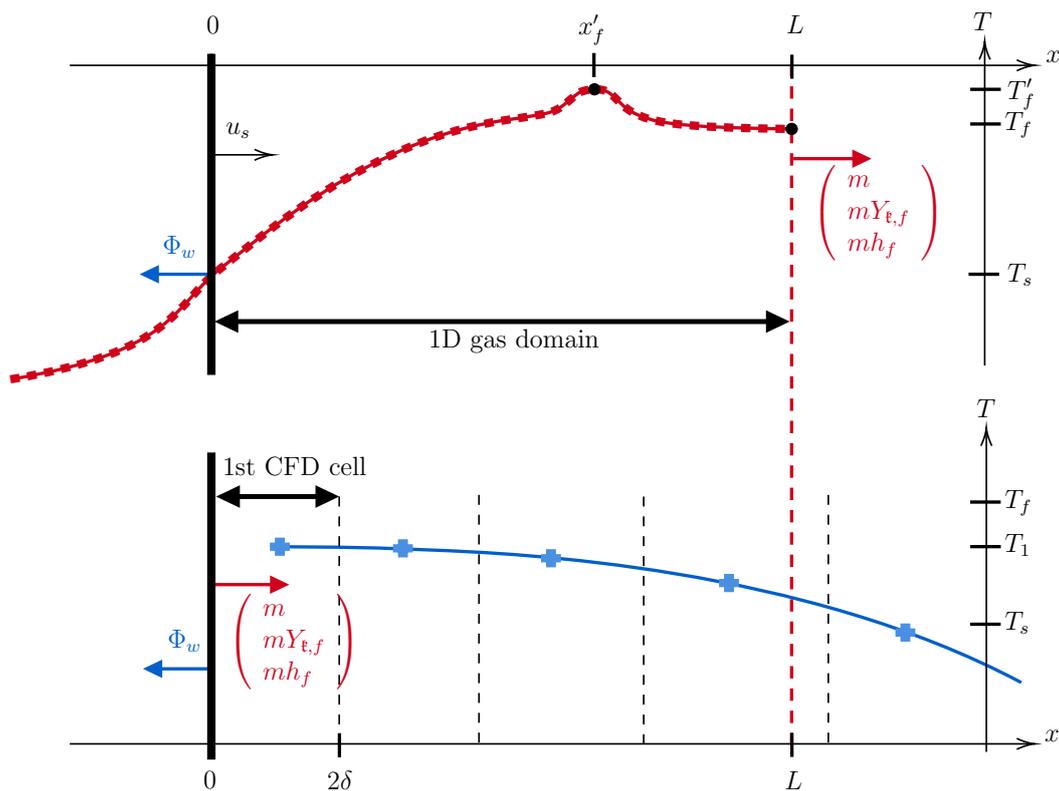
Several conceptual issues arise from this overlap, which we discuss in the following.

### 7.3.1 Conservativity, time lag and flame quasi-steadiness

The first problem occurs if we use a fully unsteady 1D gas phase in the boundary model. In that case, a change in the parietal heat flux  $\Phi_w$  or pressure  $P_w$  provided by the CFD solver has an instantaneous effect on the interface in the 1D model and on the 1D gas phase, because the 1D gas thermodynamic pressure field is uniform and equal to the parietal pressure, and the wall heat flux is directly applied at the propellant surface in the 1D model (blue arrows in Figure 7.4). However, the resulting perturbation of the 1D model will need to be convected from the surface of the propellant to the exit of the 1D gas domain. Thus, if we inject in the CFD domain the fluxes that exit the 1D gas phase, the result of the input perturbation will be injected after a delay on the order of  $L/u_s$ , with  $u_s$  the 1D gas flow velocity at the surface. An arbitrary and non-physical delay results.

This issue has also been clearly identified in some early chamber ignition simulations we performed. Initially, both the combustion chamber domain and the 1D flame domains are filled with inert cold gases. As ignition occurs following the impingement of hot igniter gases on the propellant surface, a strong mass injection establishes itself at the exit of the 1D domains. However, if the 1D flame is solved in an unsteady manner, this strong flow rate will first expel the cold gases initially present in the 1D gas domain, resulting in an important and unphysical cold gas injection in the CFD domain. The duration of that cold injection is also directly proportional to the length of the 1D gas domain.

Ad hoc corrections can be thought of, e.g. injecting in the CFD domain the fluxes obtained at a dynamically calculated flame height  $x'_f$  in the 1D domain, for instance the point of highest temperature ( $x'_f$  in Figure 7.4). However, conservativity of the coupling is lost, because these fluxes



**Figure 7.4** Illustration of the spatial coupling between the 1D code (upper part) and the CFD code (lower part), showing the overlap of the 1D flame and the CFD domain

are not equal to those that exit the 1D domain. Also there remains a convective delay on the order of  $x'_f/u_s$ , which may still be erroneous.

This problematic would not occur if the 1D domain only extended up to the center of the first CFD cell above the surface. Then, direct connection between both domains could be possible by imposing the CFD state in that cell as the state at the end of the last 1D gas cell. However, with fine meshes required for an accurate computation of the conjugate heat transfer (on the order of 1 to 10  $\mu\text{m}$  [58]), the propellant flame would not be complete within the 1D domain, and would therefore need to be solved within the CFD solver as well, adding further complexities and computational cost to the overall coupled framework.

An alternative is to enforce the quasi-steadiness of the 1D gas phase. Thus, the flame instantaneously adapts itself to the variations of the input provided by the CFD solver, without any delay. Conservativity is ensured by injecting in the CFD domain the fluxes that exit the 1D domain. The downside of this approach is the loss of dynamic effects resulting from the unsteady character of the 1D flame. However little is known experimentally about such effects, owing to the difficulty of obtaining precise measurements of the combustion response to high-frequency pressure oscillations.

The quasi-steady gas phase model thus corresponds to the classical QSHOD assumption (Quasi-Steady Homogeneous One-Dimensional) used for studying the response of solid propellants to pressure oscillations [156, 190]. This is also in line with the previous ignition models from ONERA [51], where the gas flame is assumed quasi-steady, but modelled in a simpler manner that allows for an analytical solution to be derived. From a mathematical point of view, the whole gas phase system (7.28) to (7.30) becomes algebraic of index-1, hence the specific time schemes presented in Chapter 5 are still appropriate.

Note that, before ignition, the surface mass fractions of the reactants are negligible, thus no reaction occur and the quasi-steady gas phase is uniform, with a temperature, composition and mass flow rate equal to those found at the surface.

### 7.3.2 Transition of the wall heat flux during ignition

Another non-trivial aspect is the formulation of the wall heat flux  $\Phi_w$  transmitted by the CFD solver to the propellant solver. It appears in the surface energy balance Equation (7.33), which we can reformulate in the simpler form, in the same spirit as Equation (2.27):

$$(\lambda_c \partial_\eta T)_{0^-} = (\lambda \partial_\eta T)_{0^+} + m Q_p + \Phi_w \quad (7.35)$$

It is important to clarify the significance of this equation. The surface processes and the heating of the solid phase are powered by two sources: the 1D propellant flame, and the thermal layer above the surface in the CFD domain. However, both sources cannot be simply combined. To demonstrate this difficulty, we now analyse the different situations which can be encountered. A propellant boundary face of the CFD domain successively goes through the following stages: inert heating, ignition, established burning. Let us analyse each of these phases in details.

#### Inert heating

When the propellant is not ignited, it behaves as an inert wall which is heated up by the convective heat flux  $\Phi_w$  induced by the hot gas flow over its surface. This effect cannot be captured by the 1D flame model, since it does not represent the outer flow. Only the chamber fluid model of the CFD code is able to provide the value of this heat flux. This is the reason why this wall heat flux is imposed directly at the surface of the solid propellant, as described by Equation (7.33) and Figure 7.4.

Thus, it bypasses the 1D gas phase, which may seem surprising at first. In our case, this gas phase is anyway enforced to behave in a quasi-steady manner, hence, before ignition, all the fields from the 1D gas phase are uniform and equal to their values at the surface, for instance  $T = T_s$  across the whole 1D gas domain. Consequently, the conductive heat flux produced by the 1D flame is  $(\lambda \partial_\eta T)_{0^+} = 0$ . It does not have any effect on that initial heating period, in particular the 1D flame model does not take heat away from the propellant surface via heat diffusion. That effect can however be captured with the chamber fluid model, as will be demonstrated later in Section 7.4.

To evaluate the wall heat flux, we use a first-order finite difference approximation of the temperature gradient along the normal to the propellant surface:

$$\Phi_w = (\lambda \partial_\eta T)_w = \lambda \frac{T_1 - T_s}{\delta/2} \quad (7.36)$$

where  $T_1$  is the gas temperature of the chamber fluid model at the center of the first CFD cell above the propellant surface point considered, and  $\delta$  is the height of that cell. Note that for coarse CFD meshes, wall laws could be used to improve the precision of this formula by applying a corrective factor [245].

#### Established burning

Once the propellant is ignited, we assume that the 1D propellant flame completely isolates the propellant from the surrounding flow. The only influence that the chamber fluid model from CHARME

can exert on the propellant combustion model from VULC1D is through the pressure  $P$ . The only source of heat at the propellant surface is the conductive heat feedback from the 1D flame, therefore the wall heat flux  $\Phi_w$  is set to 0.

## Ignition

Between the two previous phases, ignition occurs. During that short period, the solid propellant transitions from an inert material to an unsteady burning material with a large injected mass flow rate. Hence, there is also a transition that occurs in terms of conjugate heat-transfer. As we just discussed, after ignition the propellant flame is expected to be the sole source of thermal energy for the propellant, and its blowing effect supposedly isolates it from the surrounding flow. Therefore it is reasonable to formulate the wall heat flux such that it becomes zero at this stage.

Using the inert heating formulation (7.36) after ignition results in a very large “fictive” heat flux, because  $T_1$  tends to the flame temperature  $T_f$ , which can be several times higher than  $T_s$ . This greatly enhances the propellant consumption and should therefore be avoided. Moreover this post-ignition wall heat flux is proportional to  $(T_1 - T_s)/\delta \approx (T_f - T_s)/\delta$ . The numerator is independent of  $\delta$ , hence this expression diverges as the mesh is refined.

Another approach is to take the 1D flame temperature  $T_f$  as the wall temperature  $T_w$ :

$$\Phi_w = \lambda \frac{T_1 - T_f}{\delta/2} \quad (7.37)$$

with  $T_f$  the flame temperature (exit temperature from the 1D gas phase). The underlying idea is that  $T_f = T_s$  during the inert heating period, and that  $T_1$  tends to  $T_f$  once ignition has occurred, thus the heat flux will gradually decrease during ignition. Still, numerical experiments have shown that the conductive heat flux from Equation (7.37) ends up being slightly negative during established burning, which is physically debatable. More importantly, during the ignition period  $T_f$  will rise very rapidly, much more quickly than  $T_1$ . Consequently  $\Phi_w$  may be very large and negative during this transition period, causing an artificial slowdown of the ignition. In addition, this cooling flux also diverges as the mesh is refined, hence it is not acceptable.

A first correction could be to drive the wall heat flux to 0 by multiplying it by a sigmoid  $\sigma$ :

$$\Phi_w = \lambda \frac{T_1 - T_s}{\delta/2} \sigma(T_f) \quad (7.38)$$

For example,  $\sigma$  can be such that for  $T_f < 2300$  K,  $\Phi_w$  is unaltered, and for  $T_f > 2700$  K,  $\Phi_w$  is zero, with a smooth transition between both points. Typically, this could be achieved with  $\sigma(T_f) = 0.5 \left( 1 - \tanh\left(\frac{T_f - 2500}{100}\right) \right)$ . The sigmoid could also be applied on  $m$ , the surface mass flux instead. Still, this pragmatic approach is artificial and different choices of the transition parameters may influence the ignition dynamics.

An alternative is to choose the same smoothing factor as for the heat transfer coefficient obtained in steady-state boundary layers with parietal mass injection [257], where the heat flux is multiplied by a decreasing exponential of the mass flow rate. This approach does however not seem suitable for the highly unsteady ignition event which we aim at simulating. Moreover, this correction assumes that the boundary layer above the propellant (which is a reactive layer in reality) behaves similarly to a conventional inert boundary layer with transpiration. This requires further investigation [257, 258].

## Hybrid formula

In the present manuscript, to circumvent the issues mentioned in the previous paragraphs, we introduce and use the following hybrid formula:

$$\Phi_w = \frac{\lambda}{\delta/2} \min \left( \max \left[ \min (T_1 - T_s, T_1 - T_f), 0 \right], T_1 - T_s \right) \quad (7.39)$$

To understand this formula, it is useful to consider once again the 3 phases we have previously analysed. First, during the inert heating period, we have  $T_s = T_f < T_1$ . Thus  $\Phi_w$  is identical to Equation (7.36). During ignition,  $T_s < T_f$ , thus the hybrid formula becomes identical to Equation (7.37), the only difference being that once  $T_f$  has become larger than  $T_1$ , the heat flux is set to 0 instead of becoming negative. After ignition,  $T_f \geq T_1$  thus  $\Phi_w = 0$ . The last case covered by the hybrid formula is the cooling of an unignited propellant. If the propellant has been locally heated up without reaching ignition, and the parietal flow suddenly becomes colder, then  $T_1 < T_s = T_f$ , thus Equation (7.39) yields  $\Phi_w = \frac{\lambda}{\delta/2}(T_1 - T_s) < 0$ , effectively cooling down the propellant surface. The heat flux transitions that occur during ignition are still mesh-dependent, however this dependence is largely limited by the hybrid formula.

The evolutions of  $T_1$ ,  $T_s$ ,  $T_f$  for one surface point in a simulation from Chapter 11 is shown in Figure 7.5a. The corresponding evolutions of the various wall heat flux formulations discussed in this section are plotted in Figure 7.5b. Note that the data comes from a simulation that used the hybrid formulation from Equation (7.39). The evolutions of the various temperatures would change if other flux formulations were used, however this comparison is still very representative. Figure 7.5c shows the evolution of the various terms appearing in Equation (7.35). We clearly see that the rate of energy absorption by the solid phase,  $(\lambda_c \partial_\eta T)_{0-}$ , is dominated by the convective heat transfer with the fluid model through  $\Phi_w$  before ignition. After ignition, the only source of energy for the solid phase heating and the surface processes is the 1D flame heat feedback  $(\lambda \partial_\eta T)_{0+}$ .

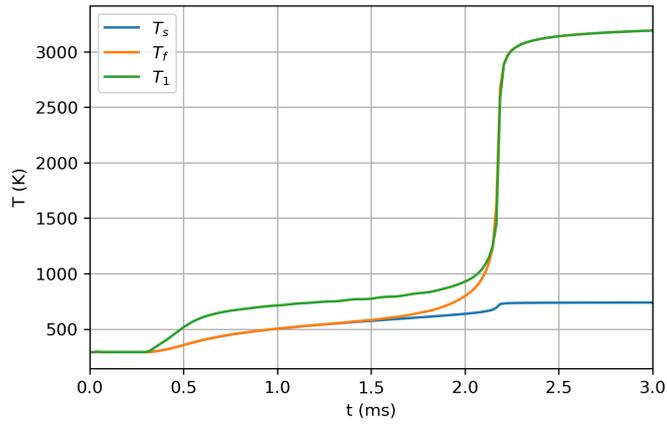
To our knowledge, all other ignition models from the literature behave like switches. The transition between an inert heating behaviour and a quasi-steady burning is instantaneous, typically upon reaching a predefined ignition temperature. Hence, the critical ignition interval where the propellant transitions from inert heating to established burning does not appear in these simulations.

## 7.4 One-dimensional verification

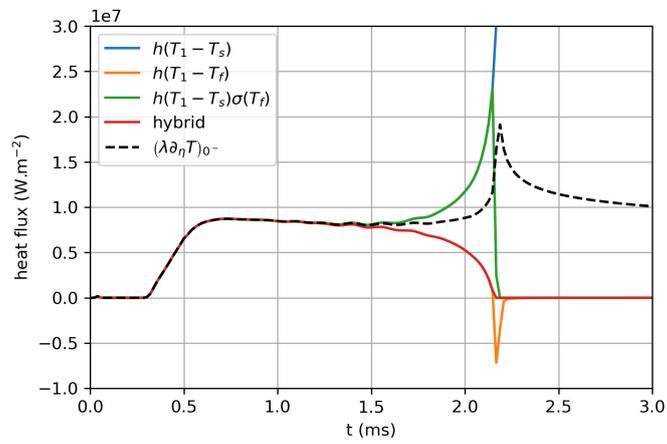
We now verify the implementation of the previous models in a one-dimensional case of laser-induced ignition, i.e. with an additional constant heat flux imposed at the propellant surface. We use the propellant model that has been obtained via an optimisation procedure which will be presented in Chapter 9, in particular the parameters from Section 9.4.

The solid propellant solver VULC1D has already been verified in Chapter 5 for the laser-induced ignition of a propellant sample. Thanks to its time adaptation capability and high-order temporal schemes, as well as the use of a highly-refined 1D mesh, the obtained solution is very precise and can be used as a reference. The previously presented coupled approaches can then be tested with a similar physical configuration to verify their ability to reproduce the one-dimensional ignition, using CHARME to solve the gas phase dynamics.

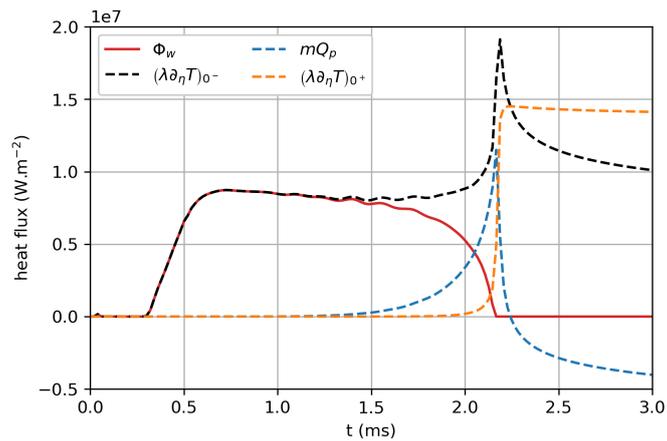
Both the standalone VULC1D code and the coupled framework are parametrised with the previous model parameters. The gas phase is at 5 bars, and the initial temperature field is uniform at  $T = 300$  K, both in the solid and gas phases. The gas is initially composed only of combustion



(a) Evolution of  $T_1$ ,  $T_s$ ,  $T_f$



(b) Effect of the wall heat flux formulation

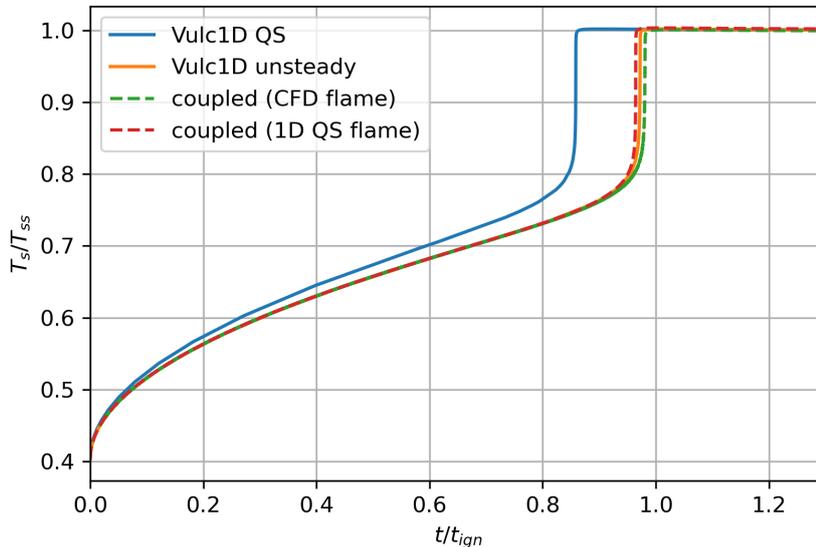


(c) Evolutions of the terms involved in the surface energy balance (7.35)

**Figure 7.5** Typical surface evolution for a single propellant point during a simulated chamber ignition

products  $G_2$  that act as the initial pressurisation inert gas. For the coupled approaches, the CFD mesh has cells with a geometrical progression in thickness. For the 1D coupled flame approach, the size of the first cell above the surface is  $10\ \mu\text{m}$ , which is sufficient to accurately capture the heat loss via diffusion through the gas phase during the initial heating of the surface. For the detailed coupled approach, the proper resolution of the propellant flame in the CFD domain requires a finer mesh, thus the first cell is reduced to  $1\ \mu\text{m}$ .

At  $t = 0\ \text{s}$ , a  $1\ \text{MW}\cdot\text{m}^{-2}$  laser heat flux is applied at the propellant surface via an additional heat flux, inserted in the surface balance equation (7.21) as previously done for  $\Phi_w$ . The surface temperature rises, some energy being lost via diffusive heat transfer to the gas phase. At one point, the pyrolysis mass flow rate becomes important and a sufficient amount of gaseous reactants is expelled from the surface, forming a flame that attaches to the surface, causing a rapid increase in surface temperature. Finally, the system converges to steady state. The obtained surface temperature evolutions are shown in Figure 7.6.



**Figure 7.6** Comparison of the surface temperature evolutions for the laser-induced ignition

The reference result is the orange curve, which is obtained with VULC1D only, solving both the unsteady solid and unsteady gas phase with a fifth-order adaptive time scheme. The dashed curves are obtained with the two coupled approaches, which are first-order in time. The green dashed curve is obtained with the detailed coupled approach. The ignition time is 0.9% larger than the reference result, which may be attributed to the coarser gas mesh used in the coupled approach. The red dashed curve is obtained with the 1D flame coupled approach, with a quasi-steady flame. The heat loss at the surface via diffusion to the CFD gas phase is correctly captured, and ignition occurs 0.8% faster than in the reference simulation. This may be attributed to the faster appearance of the propellant flame, following the quasi-steady assumption (QS).

We have also added the blue curve of  $T_s$  obtained with VULC1D only and a quasi-steady gas phase, using a fifth-order adaptive time scheme. We observe that, before ignition, the temperature rises more quickly. This is due to the lack of diffusive heat loss to the gas phase, as the quasi-steady flame model results in a uniform gas temperature field at  $T = T_s$  before ignition. Therefore, the point of ignition is reached more quickly, in this case 12 % faster than in the reference simulation.

Consequently, the quasi-steady flame assumption used in the 1D-flame coupled approach does not modify much the overall ignition behaviour. Indeed, the heat loss by diffusion to the gas phase can still be captured by the direct connection between the fluid and propellant models via the wall heat flux  $\Phi_w$  in Equation (7.33). However, note that the flame appearance is faster in quasi-steady mode, so there might be a more visible difference in a configuration where the establishment time of the unsteady flame is not small compared to the characteristic time of ignition.

Overall, we see that both coupled approaches produce nearly identical results for the one-dimensional laser-induced ignition, and that the agreement with the higher-precision standalone VULC1D code is excellent. This verifies the coupled approaches presented in this chapter as well as their implementations.

## Partial conclusion

In this chapter, we have described the fluid model used for the simulation of the chamber flow field. The connection conditions required for the representation of the interaction with the propellant model have been presented, first in the case of the detailed approach, where the propellant flame is solved within the fluid model. Since the spatial scales associated with the propellant flame are small, it is sensible to consider a 1D model of the latter, as in Chapter 2. This model is introduced as an additional layer between the solid phase model and the fluid model.

From a technical point of view, the simulation of the interaction between these models is performed by coupling the code VULC1D presented in Part I and the CFD tool CEDRE from ONERA. A simple staggered coupling algorithm is described, with first-order accuracy in time.

Several issues arising from the one-dimensional flame model are discussed. In particular, a conservative and temporally coherent coupling can only be achieved by enforcing the quasi-steadiness of the propellant flame in the 1D flame coupled approach. Also, concentrating the effect of the propellant flame at the surface of the propellant for the fluid model brings a number of complications for the conjugate heat transfer.

To verify the proposed coupled models and their implementations, a comparison of the 1D flame coupled approach and the detailed coupling has been presented for the laser-induced ignition of a solid propellant in 1D. Both coupled approaches yield results similar to those of VULC1D.



# Chapter 8

## Towards a higher-order adaptive coupling

### Summary

Different approaches to increase the order of convergence in time of the coupling between CEDRE and VULC1D are described. Exploratory work conducted on the use of polynomial extrapolation in time of the coupling variables is presented, with applications to simplified test cases. This method shows interesting potential for dynamic time step and order adaptation, which should be further investigated in future work.

In the previous chapter, we have described a coupling algorithm which is first-order accurate in time, and where the coupling time step is dictated by the CFD solver. It is known that low-order of convergence in time can strongly deteriorate the dynamics of a system, which may quickly drift away from the exact solution's dynamics. This has clearly been demonstrated in Chapter 6, where high-order (order 3 and above) integration schemes performed much better than lower-order ones, both in terms of accuracy (for a given time step) and computational efficiency (for a given error tolerance). As discussed in the general introduction, this issue is also found in all SRM ignition simulations reported in the literature.

It would be advantageous to have a higher-order approach, with the additional ability of automatically determining the coupling time step based on objective error estimates, in a similar manner as for the embedded method approach from Chapter 5. Therefore, the objective of this chapter is to find a technique which can be implemented simply in the existing CEDRE framework. In this chapter, we report on an initial exploration of different techniques to increase the order of our original coupling algorithm.

Let us consider the following generic and simple form of our problem, where the solid and fluid domains are associated with the subscript 1 and 2 respectively:

$$d_t y_1 = f_1(y_1, y_2) \quad (8.1)$$

$$d_t y_2 = f_2(y_1, y_2) \quad (8.2)$$

with  $y_i \in \mathbb{R}^{n_i}$  denoting the state vector of each solver, and  $n_i$  the dimension of the  $i$ -th subsystem. We assume appropriate initial conditions are specified. Let us investigate some of the possibilities that are available for a high-order integration of such a system.

**Monolithic implicit approach** A first idea would be to integrate the previous system in a monolithic manner, applying the same integration scheme to both the 1D propellant models and

the CFD fluid model. For instance, the  $i$ -th stage of a DIRK method applied to both systems would read:

$$\underbrace{\begin{pmatrix} y_{1,ni} \\ y_{2,ni} \end{pmatrix}}_Y = \underbrace{\begin{pmatrix} y_{1,n} + \Delta t \sum_{j=0}^i a_{ij} f_1(y_{1,nj}, y_{2,nj}) \\ y_{2,n} + \Delta t \sum_{j=0}^i a_{ij} f_2(y_{1,nj}, y_{2,nj}) \end{pmatrix}}_{G(Y)} \quad (8.3)$$

where  $Y = (y_{1,ni}^t, y_{2,ni}^t)^t \in \mathbb{R}^{n_t}$ , with  $n_t = n_1 + n_2$ .

Note that the index-1 variables present in our 1D propellant model can be embedded as functions of the state vector, following the state-space form introduced in Chapter 5. Thus, at each stage a nonlinear problem  $0 = F(Y)$  needs to be solved, where the operator  $F$  is defined by  $F(Y) = Y - G(Y)$ . Applying a Newton method on the whole system is complex and rarely done in practice, even though research has been conducted on performing Newton iterations directly on coupled systems originally solved by separate solvers [259], by exchanging Jacobians of the coupling terms, or by performing global Jacobian-free Newton-Krylov iterations. This however requires access and modifications to the code of both solvers.

Alternatively, a fixed-point approach can be used, where multiple simple iterations are performed, starting from an initial guess  $Y^0$  of the solution. The  $(k + 1)$ -th iteration is defined as:

$$Y^{k+1} = G(Y^k) \quad (8.4)$$

Under the condition that  $G$  is Lipschitz-continuous with a Lipschitz constant  $L_g < 1$ , i.e.:

$$\|G(a) - G(b)\| < \|a - b\| \quad (8.5)$$

with  $a, b \in \mathbb{R}^{n_t}$ , there exists a unique fixed-point  $Y^*$  such that  $Y^* = G(Y^*)$ , and the fixed-point iterations converge towards it. The iterations can be stopped once the variations of  $Y$  from one iteration to the next are small enough:

$$\|Y^{k+1} - Y^k\| < tol \quad (8.6)$$

For a sufficiently low tolerance  $tol$ , the solution can be considered identical to the solution  $Y^*$  obtained with a hypothetical monolithic solver that would solve both models in a fully coupled manner with the same integration scheme. If the functions  $f_i$  are Lipschitz-continuous with Lipschitz constants  $L_{f_i}$ , then the contractivity condition on  $G$  is ensured if  $\Delta t < 1/(a_{ij} \max(L_{f_i}))$ , i.e. the fixed-point process is ensured to converge if the time step is small enough. The convergence is linear, i.e. the error  $\|Y - Y^*\|$  is decreased by a factor  $L_g$  at each iteration. Hence, a smaller time step is preferable. In particular, the stability of the fixed-point-iterated scheme is the same as the explicit scheme that produces the first iteration, hence A-stability cannot be obtained. In the case of a stiff coupling, the iteration may diverge. This difficulty can be overcome by solving the nonlinear system (8.3) with a Newton method instead [260]. This fixed-point approach is the fundamental idea of *waveform relaxation* (WR), which aims at solving problems involving separate systems [254] or multiple space and/or time domains [261, 262]. WR is more general, insofar as waveforms can be iteratively exchanged (elements of a function space) rather than discrete variables.

The fixed-point iteration (8.4) for the system (8.3) reads:

$$y_{1,ni}^{k+1} = y_{1,n} + \Delta t \sum_{j=0}^i a_{ij} f_1(y_{1,nj}^{k+1}, y_{2,nj}^k) \quad (8.7)$$

$$y_{2,ni}^{k+1} = y_{2,n} + \Delta t \sum_{j=0}^i a_{ij} f_2(y_{1,nj}^k, y_{2,nj}^{k+1}) \quad (8.8)$$

This is classically referred to as a Jacobi iteration, by analogy with the Jacobi algorithm for the solution of linear systems. Continuing with the same analogy, another form is the Gauß-Seidel iteration:

$$y_{1,ni}^{k+1} = y_{1,n} + \Delta t \sum_{j=0}^i a_{ij} f_1(y_{1,nj}^{k+1}, y_{2,nj}^k) \quad (8.9)$$

$$y_{2,ni}^{k+1} = y_{2,n} + \Delta t \sum_{j=0}^i a_{ij} f_2(y_{1,nj}^{k+1}, y_{2,nj}^{k+1}) \quad (8.10)$$

Here, Equation (8.9) is solved first to provide the value of  $y_{1,ni}^{k+1}$  for the solution of Equation (8.10). This iteration scheme usually improves the convergence rate, however it is inherently sequential, therefore less efficient in some applications. Note that the coupling algorithm from the previous chapter, in particular Figure 7.3, corresponds to a single Gauß-Seidel iteration of the waveform relaxation procedure.

In the case of conjugate heat transfer between an inert solid and a compressible gas phase, this fixed-point approach has been reported as an effective method in the case of strongly coupled systems [263–265]. In these studies, the solvers use the same temporal integration scheme and exchange their solutions multiple times at each Runge-Kutta stage to perform the WR procedure.

**Implicit-explicit approach** To avoid the cost of iterating at each stage of the Runge-Kutta method, an explicit method could be applied, so that a single fixed-point iteration is sufficient, since  $F$  does not depend on  $Y_{ni}$  in that case. However, applying an explicit scheme to both fluid and solid models results in a very poor stability, in particular with highly refined meshes. An intermediate approach has been proposed in [266], where an implicit-explicit (IMEX) approach is used. Both solid and fluid solvers simulate their own dynamics with the same implicit scheme, however the coupling terms are handled via a “companion” explicit scheme, whose coefficients are derived from the ones of the implicit scheme based on order conditions. Thus, at each stage of the Runge-Kutta scheme, the unknowns of the overall system state vector can be determined by each solver separately. They are then exchanged (via the coupling terms) at the next stage of the method. This partial explicitation of the integration may however hurt the overall stability of the system, in particular in the case of fast dynamics of the interface coupling both models.

Such quasi-monolithic approaches do not allow for temporal substepping inside each solver, which may result in inefficiencies if the time scales of the fluid and solid models differ largely. Furthermore, in the case of our coupling of VULC1D and CEDRE, these approaches are not adapted to the technical choices already made in the development of CEDRE, which assume separate time integration of all coupled solvers. In particular, the CFD solver CHARME is especially designed for linearised schemes, and the ability to iterate on a given time step (i.e. restart a step) is not adequately supported. Hence we cannot consider such a monolithic approach.

There also exist semi-implicit (or linearly implicit) approaches [267], where a single Newton-like iteration is performed, based on a local linearisation of the coupled problem. This semi-implicit approach yields a stability intermediate between explicit and implicit schemes, without requiring an iterative procedure. However, it requires the knowledge of the Jacobians  $\partial_{y_j} y_i$ , which indicate how the solution of the  $i$ -th system at time  $t_{n+1}$  depends on the solution of the  $j$ -th system. In practice,

only a part of  $y_j$  directly influences the evolution of  $y_i$  (e.g. near-surface variables in our case), thus greatly lowering the number of partial derivatives to be computed. Forming these Jacobians may however involve a costly procedure, where a coupling step is performed multiple times with perturbed solution values to approximate these Jacobians by finite-differences. Consequently, we do not investigate this approach further.

**Operator splitting** We have already noted that the coupling algorithm from the previous chapter corresponds to a single Gauß-Seidel iteration of the waveform relaxation procedure. In the case of a single-stage Runge-Kutta method, e.g. backward Euler, it may also be interpreted as a Lie splitting of the system:

$$\frac{\partial}{\partial t} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} f_1(y_1, y_2) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ f_2(y_1, y_2) \end{pmatrix} \quad (8.11)$$

with system 1 being integrated first. It is preferable to integrate the stiffest operator first to avoid order reduction phenomena [268]. We can extend that to the use of any integration schemes. Let us assume each subsystem has an integrator of the form:

$$y_{i,n+1} = y_{i,n} + \Delta t F_i(y_{1,n}, y_{2,n}; \Delta t) \quad (8.12)$$

Then, the Lie splitting reads:

$$y_{1,n+1} = y_{1,n} + \Delta t F_1(y_{1,n}, y_{2,n}; \Delta t) \quad (8.13)$$

$$y_{2,n+1} = y_{2,n} + \Delta t F_2(y_{1,n+1}, y_{2,n}; \Delta t) \quad (8.14)$$

As long as the algorithms  $F_1$  and  $F_2$  used to integrate each subsystem are consistent (i.e. of order at least 1), the Lie splitting is of order 1, which is coherent with the discussion from the previous chapter. A natural improvement is to transform this to a Strang splitting. The iterations (8.9) and (8.10) are then replaced by:

$$y_{1,n+\frac{1}{2}} = y_{1,n} + \frac{\Delta t}{2} F_1 \left( y_{1,n}, y_{2,n}; \frac{\Delta t}{2} \right) \quad (8.15)$$

$$y_{2,n+1} = y_{2,n} + \Delta t F_2 \left( y_{1,n+\frac{1}{2}}, y_{2,n}; \Delta t \right) \quad (8.16)$$

$$y_{1,n+1} = y_{1,n+\frac{1}{2}} + \frac{\Delta t}{2} F_1 \left( y_{1,n+\frac{1}{2}}, y_{2,n+1}; \frac{\Delta t}{2} \right) \quad (8.17)$$

The obtained splitting is of order 2 [269]. A technical advantage of this formula becomes apparent if we consider two consecutive time steps, from  $n$  to  $n+1$ , and  $n+1$  to  $n+2$ . By combining Equation (8.17) for the first one and Equation (8.15) for the second one, we obtain the following system:

$$y_{1,n+\frac{3}{2}} = y_{1,n+\frac{1}{2}} + \Delta t F_1 \left( y_{1,n+\frac{1}{2}}, y_{2,n+1}; \Delta t \right) \quad (8.18)$$

$$y_{2,n+2} = y_{2,n+1} + \Delta t F_2 \left( y_{1,n+\frac{3}{2}}, y_{2,n+1}; \Delta t \right) \quad (8.19)$$

This is identical to a Lie splitting from time  $t_{n+1}$  to time  $t_{n+2}$ , with  $y_1$  shifted by half a step backward. Thus a complete integration of a coupled system with a Strang splitting is technically equivalent to a Lie splitting, with the exception that half a step of the first subsystem have to be performed before the first time step and after the last step.

Generalisation to higher-order is more complex, since it involves negative or complex-valued splitting time steps [270]. Also, derivation of an error estimate to dynamically adapt the splitting time step is more involved [234, 235]. Therefore, we discard this approach.

**Polynomial extrapolation of the coupling variables** In the basic coupling algorithm described in the previous chapter, the solvers do not exchange their complete solution vectors  $y_i$ . Instead, only a subset of these vectors are exchanged (e.g. near-surface variables), or even a function of such a subset (e.g. surface temperature, fluxes). Let us denote the values transmitted to the  $i$ -th solver as  $u_i$ , which we will refer to as inputs, or coupling variables. In this basic coupling procedure, these inputs are updated at the beginning of a coupling step and maintained constant during that step. Obviously, the error in the coupling variables is of first order in time during the integration of each submodel. An improvement on that could be to assume a linear variation in time of these variables instead. Such a linear extrapolation could be constructed in a simple manner by using the two previous coupling points:

$$\hat{u}_i(t) = u_{i,n} + \frac{u_{i,n} - u_{i,n-1}}{t_n - t_{n-1}}(t - t_n) \quad (8.20)$$

This procedure can be extended to more refined evolutions by relying on a larger number ( $p_i + 1$ ) of previous steps and constructing a polynomial extrapolation of the coupling variables. Thus, for the time interval  $[t_n, t_{n+1}]$ , the input  $u_i$  is replaced by the following Newton formula:

$$\hat{u}_i(t) = u_{i,n} + \sum_{j=1}^{p_i} \delta^j u_i[t_n, \dots, t_{n-j}] \prod_{k=0}^{j-1} (t - t_{n-k}) \quad (8.21)$$

where  $\delta^j u_i$  denotes the  $j$ -th divided difference of  $u_i$ . They are defined by the following recurrence formula:

$$\delta^j u_i[t_n, \dots, t_{n-j}] = \frac{\delta^{j-1} u_i[t_{n-1}, \dots, t_{n-j}] - \delta^{j-1} u_i[t_n, \dots, t_{n-j+1}]}{t_{n-j} - t_n} \quad (8.22)$$

and the initial value  $\delta^0 u_i[t_n] = u_{i,n}$ .

This approach is similar to the principles of the backward differentiation formula (BDF) for the solution of ODEs [218]. Note that the Newton formula may be numerically unstable as it is poorly conditioned, however this issue only appears for a large number of sampling points (typically more than 10) and will not be an issue in this chapter.

This polynomial extrapolation is the central idea of *co-simulation*, a domain which is focused on the coupling of multiple solvers for transient simulations. Dynamic adaptation of the coupling time step and demonstrations of high-order convergence in time have been reported [267, 271, 272]. Based on the technical framework of CEDRE, this approach seems to be well-suited to our needs, both in terms of versatility and ease of implementation.

Therefore, in this chapter, we first present an overview of the co-simulation framework. In particular, we show how the extrapolation impacts the precision of the coupled simulation. Then, a representative one-dimensional toy model of conjugate heat transfer with a reactive interface is presented, and we demonstrate that high orders of convergence can be obtained. Then, we show on a simple SRM model how this approach can be used to automatically drive the coupling time step and the order of accuracy, so as to improve the computational efficiency of a coupling and its stability.

## 8.1 Co-simulation

Co-simulation is a domain that appeared in the 1990s and has become very active since the 2000s. Extensive research has been produced, as recently reviewed in [273]. Its main focus is on the simulation of multidisciplinary systems, involving a combination of subjects, e.g. mechanics, electronics,

hydraulics, thermodynamics, each of which already possesses a specific solver. Reimplementing all the separate subsystem solvers as a complete system is usually inconceivable for multiple reasons:

- Each solver may use very different approaches (finite element, rigid body mechanics, finite volume...) and benefits from specialized solution procedures (e.g. adequate time integration methods). A monolithic solver would sacrifice optimality and speed in order to cope with the various methods that need interfacing, as well as the potentially vastly different time scales involved in the subsystems.
- Each solver may consider very different physics (mechanics, fluid dynamics...), thus requiring expertise from many fields to be reimplemented in a complete solver. This could also result in a poor conditioning of the monolithic system.
- The solvers may be provided by third-parties and contain information that cannot be exposed, both for technical and legal reasons, thus preventing any potential remake.

Co-simulation aims at circumventing this issue by coupling the solvers dynamically and transmitting the required coupling data as input to the different solvers. It stems from the multibody simulation community [274]. In this particular domain, various mechanical systems (suspensions, tires...) possess dedicated solvers to compute their dynamics as isolated systems. In order to compute the dynamic behaviour of the complete system, these solvers must be coupled: the suspension applies a force on the wheel, from which the tire model can compute the deformation of the rubber, thus giving a new position for the suspension connection points, and so on... Actually, the subsystems are coupled via a small set of values, typically the displacement of an attachment point, or a force through a joint. That is, the coupling between the subsystems is precisely identified to a small subset of variables relative to the size of the complete system.

The idea is to step each subsystem forward in time independently and to only synchronize them at specific coupling time points. Between consecutive points, each subsystem is provided with a “guess” of the inputs from the other subsystems. The inputs are usually polynomial extrapolations of the inputs obtained at the previous coupling time steps, as we suggested in Equation (8.21).

The fundamental point of co-simulation is to have a system which can be decomposed in  $N$  multiple subsystems of the form:

$$\partial_t z_i = f_i(z_i, u_i, t) \quad (8.23)$$

$$y_i = g_i(z_i, u_i) \quad (8.24)$$

with  $z_i \in \mathbb{R}^{n_i}$  the state vector of the  $i$ -th system,  $y_i \in \mathbb{R}^{m_i}$  its output vector, and  $u_i \in \mathbb{R}^{l_i}$  its input vector. The output vector is computed as a function of the state and input vectors. To enable a coupling between the subsystems, their outputs and inputs are linked via an additional relation:

$$\begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} L_{11} & \cdots & L_{1N} \\ \vdots & \vdots & \vdots \\ L_{N1} & \cdots & L_{NN} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \quad (8.25)$$

where the submatrices  $L_{ij}$  belong to  $\mathbb{R}^{m_i \times l_j}$  and only contain zeros or ones. Typically, a single element is non-zero in each row of  $L_{ij}$ .

Let us briefly consider what the inputs and outputs are in the case of the coupling from Chapter 7. The first system (VULC1D) receives as input  $u_1$  the pressure, temperature (and mass fractions for

the detailed approach) at the center of the first CFD cell above the propellant surface. Its output is  $y_1$ , which contains the fluxes of mass, species and enthalpy. The second system (CHARME) receives  $u_2$  as input, which is exactly equal to  $y_1$  (fluxes entering the first CFD cell). Its outputs are the values of the flow field variables at the center of that cell, i.e.  $y_2 = u_1$ .

### 8.1.1 Integration algorithm

#### Explicit integration

For a system of the form (8.23) and (8.24), we now describe how co-simulation enables the separate integration of the various subsystems. For the  $i$ -th subsystem, evaluating the time derivative of its state vector  $z_i$  requires the exact input  $u_i$ , which is a function of all state vectors  $z_j, j \in [1, N]$ . The root of the co-simulation approach is to replace it by an approximation  $\hat{u}_i$ , obtained with the  $i$ -th component of Equation (8.25), where the values of  $y_j, j \neq i$  are replaced by polynomial predictions  $\hat{y}_j$ .

For one co-simulation step from time  $t_n$  to  $t_{n+1}$ , each subsystem can be integrated forward in time as:

$$z_{i,n+1} = z_{i,n} + \int_{t_n}^{t_{n+1}} f_i(z_i(t), \hat{u}_i(t), t) dt \quad (8.26)$$

where the following algebraic equations are required to express the input  $\hat{u}_i$ :

$$y_i(t) = g(z_i(t), \hat{u}_i(t)) \quad (8.27)$$

$$\hat{u}_i(t) = L_{ii}y_i(t) + \sum_{j \neq i} L_{ij}\hat{y}_j(t) \quad (8.28)$$

The polynomial approximations  $\hat{y}_j$  are formulated similarly to Equation (8.21):

$$\hat{y}_j(t) = y_{j,n} + \sum_{k=1}^{p_j} \delta^k y_j[t_n, \dots, t_{n-k}] \prod_{m=0}^{k-1} (t - t_{n-m}) \quad (8.29)$$

Here, these predictions are formed from the values at the previous coupling step, hence the coupling between the subsystems is explicit.

#### Implicit co-simulation

The explicit nature of the previous coupling can hinder the stability of the coupled integration. A truly implicit coupling would require that the actual values of the outputs  $y_j$  be used instead of predictions. This can be approached iteratively by performing the previous integration step multiple times (referred to as iterations), each time updating the polynomial approximation so that it is an interpolation of the output evolution obtained during the previous iteration.

Denoting with  $\alpha$  the iteration number for the current coupling time step from  $t_n$  to  $t_{n+1}$ , the implicit approach can be expressed similarly to the explicit one. The first iteration is the explicit integration from the previous paragraphs. For the  $\alpha$ -th iteration ( $\alpha > 1$ ), the integration of the state vector variation (8.26) and the algebraic equations (8.27) and (8.28) only need slight modifications to introduce the iteration number  $\alpha$ :

$$z_{i,n+1}^\alpha = z_{i,n} + \int_{t_n}^{t_{n+1}} f_i(z_i^\alpha(t), \hat{u}_i^\alpha(t), t) dt \quad (8.30)$$

$$y_i^\alpha(t) = g(z_i^\alpha(t), \hat{u}_i^\alpha(t)) \quad (8.31)$$

$$\hat{u}_i^\alpha(t) = L_{ii}y_i^\alpha(t) + \sum_{j \neq i} L_{ij}\hat{y}_j^\alpha(t) \quad (8.32)$$

The iterative nature of the scheme is only apparent in the expression of the polynomial approximations  $\hat{y}_j^\alpha$  of the outputs. Instead of being extrapolations from the previous  $p_j$  coupling steps, they become interpolations from the previous  $p_j - 1$  coupling steps and the values of  $y_j$  obtained at time  $t_{n+1}$  at the previous iteration  $\alpha - 1$ :

$$\hat{y}_j^\alpha(t) = y_{j,n+1}^{\alpha-1} + \sum_{k=0}^{p_j-1} \delta^k q_j^{\alpha-1}[t_{n+1}, \dots, t_{n-k}] \prod_{m=0}^{k-1} (t - t_{n+1-m}) \quad (8.33)$$

where  $q_j^{\alpha-1}$  is defined as:

$$q_j^{\alpha-1}(t_{n-m}) = y_{j,n-m}, m \in [1, p_j - 1] \quad \text{and} \quad q_j^{\alpha-1}(t_{n+1}) = y_{j,n+1}^{\alpha-1} \quad (8.34)$$

Note that we drop the time point  $t_{n-p_j}$  so that the polynomial approximations retain the same order of accuracy.

The iterations can be stopped once the polynomial approximations have converged, i.e. when:

$$\max_{j=1..N} \|y_{j,n+1}^\alpha - y_{j,n+1}^{\alpha-1}\| < tol \quad (8.35)$$

where  $tol$  is a user-defined tolerance, typically  $10^{-10}$ . Then, the converged solution is obtained from the last iteration:  $z_{j,n+1} = z_{j,n+1}^\alpha$ .

This iterative procedure can be seen as a fixed-point algorithm. In essence, it is very close to the waveform-relaxation approach (WR) discussed in the introduction, hence it is often given the same name in the co-simulation literature. Similarly to our discussion on the monolithic approach in the introduction, two modes of the WR iterations are classically used. In the first one, the Jacobi iteration, the polynomial approximations are only updated after all the subsystems have been integrated. In the other one, the Gauß-Seidel iteration, the subsystems are integrated sequentially and the approximations are updated after each integration. Compared to the Jacobi iteration, the Gauß-Seidel iteration converges faster, however it does not allow for the subsystems to be integrated in parallel.

### 8.1.2 Local truncation error

A simplified analysis enables the determination of the dependence of the local error on the order of the approximation of the inputs. Let us assume each subsystem is associated with an exact integrator, so that the only source of error in the coupled integration is the approximation of the inputs  $u_i$ . We assume that the functions  $f_i$  are Lipschitz-continuous with respect to  $z_i$  and  $u_i$ :

$$\|f_i(z_a, u_a, t) - f_i(z_b, u_b, t)\| \leq L_{i,z} \|z_a - z_b\| + L_{i,u} \|u_a - u_b\| \quad (8.36)$$

with  $L_z$  and  $L_u$  positive real constants, and the norm  $\|\cdot\|$  is any suitable norm, e.g.  $L_\infty$ -norm.

Now, let us denote the exact solutions of the coupled problems as  $z_i(t)$  and  $u_i(t)$ , and the co-simulation solutions as  $\hat{z}_i(t)$  and  $\hat{u}_i(t)$ . We consider the time step from time  $t_n$  to  $t_{n+1} = t_n + \Delta t$ ,

with initial conditions  $\hat{z}_i(t_n) = z_i(t_n)$ . We can apply the Grönwall's lemma and obtain:

$$\|\hat{z}_i(t_{n+1}) - z_i(t_{n+1})\| \leq \underbrace{\|\hat{z}_i(t_n) - z_i(t_n)\|}_{=0} \exp(L_{i,z}\Delta t) + \frac{L_{i,u}}{L_{i,z}} (\exp(L_{i,z}\Delta t) - 1) \max_{t \in [t_n, t_{n+1}]} \|\hat{u}_i(t) - u_i(t)\| \quad (8.37)$$

Following our choice of initial conditions, the first term in the right-hand side disappears. If  $\hat{u}_i$  is a polynomial interpolation on  $[t_{n-p_i+1}, t_{n+1}]$  of  $u_i$  with degree  $p_i$ , i.e. using  $p_i + 1$  points for the Newton formula in Equation (8.21), then the approximation satisfies:

$$\hat{u}_i(t) = u_i(t) + \mathcal{O}(\Delta t^{p_i+1}) \quad t \in [t_n, t_{n+1}] \quad (8.38)$$

Thus, Equation (8.37) becomes:

$$\|\hat{z}_i(t_{n+1}) - z_i(t_{n+1})\| \leq \frac{L_{i,u}}{L_{i,z}} (\exp(L_{i,z}\Delta t) - 1) \mathcal{O}(\Delta t^{p_i+1}) \quad (8.39)$$

Considering the Taylor series of the exponential, we see that  $\exp(L_{i,z}\Delta t) - 1 = \mathcal{O}(\Delta t)$ , thus the upper bound on the error becomes:

$$\|\hat{z}_i(t_{n+1}) - z_i(t_{n+1})\| = \mathcal{O}(\Delta t^{p_i+2}) \quad (8.40)$$

Following the update of the coupling variables  $u_i$  with Equation (8.25) at time  $t_{n+1}$ , the errors of each subsystem are involved. Thus, the error on the new coupling variables  $u_{i,n+1}$  satisfies:

$$\|u_{i,n+1} - u_i(t_{n+1})\| = \mathcal{O}(\Delta t^q) \quad \text{with } q = \min_j (p_j + 2) \quad (8.41)$$

Therefore, the overall local error of the co-simulation is of order  $q$ , and the global error is of order  $q-1$ . More refined analysis of the error in co-simulations are available [267, 271, 272] with additional insights into its behaviour.

**Remark 12.** *At the end of each coupling step (time  $t_{n+1}$ ), the coupling variables are updated with Equations (8.24) and (8.25). In the case of explicit co-simulation, they generally differ from the extrapolated values  $\hat{u}(t_{n+1})$ . Therefore a small jump in the coupling variables occurs at each step and may cause an unphysical transient in the subsystems at the start of the following coupling step. Some research has been presented where the polynomial extrapolations are modified so that they are globally  $C^0$ ,  $C^1$  or  $C^2$ , and this may prove beneficial for the stability and computational efficiency of the coupling [275]. This issue has not been encountered in the test cases presented further, hence the previous corrective approach was not considered for the present work.*

**Remark 13.** *If the time derivatives of the coupling variables can be obtained, Hermite polynomials can be used instead of Lagrange/Newton [267]. With the same number of sampling points, the order of the prediction is multiplied by 2. Alternatively, a reduced number of points can be used while still maintaining the same order of accuracy. However, the time derivatives of the coupling variables are often difficult to obtain, thus we do not further investigate this possibility.*

### 8.1.3 Time step adaptation

To enable an automatic selection of the coupling time step, an estimate of this error is required. Such an estimate can be conveniently constructed by comparing the extrapolated coupling variables

and the updated ones. The true prediction error on the coupling variables is  $\hat{u}_i(t_{n+1}) - u_i(t_{n+1})$ . We can develop its norm to see how the updated coupling variables  $u_{i,n+1}$  can intervene:

$$\|\hat{u}_i(t_{n+1}) - u_i(t_{n+1})\| = \|\hat{u}_i(t_{n+1}) - (u_{i,n+1} + \mathcal{O}(\Delta t^q))\| = \underbrace{\|\hat{u}_i(t_{n+1}) - u_{i,n+1}\|}_{\epsilon_{i,n+1}^u} + \mathcal{O}(\Delta t^q)$$

For simplicity, let us assume  $p_i = p \forall i \in [1, N]$ . Then,  $q = p + 2$  and  $\|\hat{u}_i(t_{n+1}) - u_i(t_{n+1})\|$  is of the form  $\mathcal{O}(\Delta t^{p+1})$ , thus we see that  $\epsilon_{i,n+1}^u$  will converge towards the former true error. Hence, it is a reliable estimate of the prediction error, with order  $p + 1$ .

Based on this procedure, the coupling time step can be selected in a similar manner as for the embedded Runge-Kutta methods in Chapter 5. It may happen that the estimated error is too high, as discussed in Section 5.1.5. Usual solution procedures for ODEs normally would restart the current step with a smaller step size, until the error is sufficiently low. In the context of a coupling involving large-scale CFD domain, a time step restart can be too costly and is not necessarily technically possible. Therefore the step may be accepted anyway, and the time step can only be corrected afterwards. In order to better prevent the occurrence of such events, one may use a PI- or PID-controller to smoothen the time step evolution across multiple steps. This strategy is successfully adopted in [241]. Such refined controllers are widely used in co-simulation [267] and in the field of ODE/DAE integrators [206, 231, 267].

Intuitively, an error control based on the agreement between the polynomial extrapolations of the coupling variables and their updated values after a coupling time step is reassuring. Indeed, the accuracy of the predicted evolution of the coupling variables is ensured, therefore the coupled dynamics will also be correctly computed. However, it is possible that this approach is too conservative, as we cannot be certain that an error in the coupling variables will translate into a similar error in the coupled dynamics. We can imagine a situation where some coupling variables only have a weak impact on the subsystems solutions but may still vary rapidly, thus forcing a low-coupling time step to correctly capture their variations, even though the true dynamics of the coupled system is already well captured.

## 8.2 A toy model

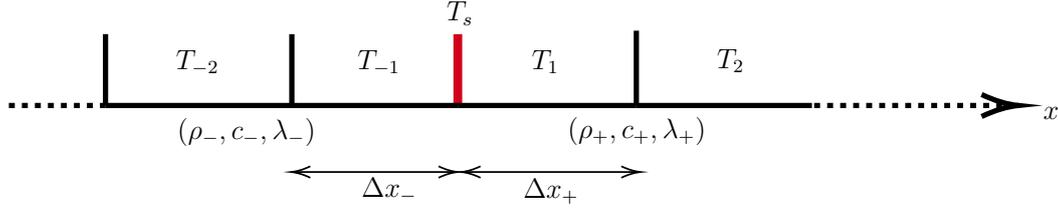
To test the co-simulation approach within a simplified framework, a simple model of the conjugate heat-transfer at the propellant surface has been designed.

### 8.2.1 Model

We consider the case of two one-dimensional solid slabs, which are inert phases where heat diffuses. They are located at  $x \in [-L_1, 0]$ ,  $x \in [0, L_2]$  respectively. We use the subscript  $-$  for the left side ( $x < 0$ ) and  $+$  for the right side ( $x > 0$ ). The two slabs are connected through an interface at  $x = 0$ , where a surface reaction generates heat. This is therefore representative of the traditional conjugate heat-transfer case, but with the addition of a reactive surface, which is closer to the propellant model from Chapter 2.

Each slab has a temperature field  $T$  subject to the following partial differential equation:

$$\rho_{\pm} c_{\pm} \partial_t T = \lambda_{\pm} \partial_{xx} T \quad (8.42)$$



**Figure 8.1** Mesh used for the toy model

with the densities  $\rho_{\pm}$ , specific heats  $c_{\pm}$  and thermal conductivities  $\lambda_{\pm}$ , which are uniform and constant in each slab.

At their outer boundary conditions, the slabs are insulated, so that the heat flux is locally zero:

$$\lambda \partial_x T(-L_1) = 0, \quad \lambda \partial_x T(L_2) = 0 \quad (8.43)$$

The interface thermal connection condition is:

$$\lambda \partial_x T(0^-) = \lambda \partial_x T(0^+) + \Phi_s(T_s) \quad (8.44)$$

where the surface reaction term has the form:

$$\Phi_s(T_s) = A \exp\left(\frac{-T_a}{T_s}\right) (T_b - T_s) \quad (8.45)$$

The form of this term allows for an initial thermal runaway to occur, after which the surface temperature stabilises at the combustion temperature  $T_b$ .

## 8.2.2 Semi-discretisation in space

For each slab, a cell-centered finite-volume scheme with a second-order approximation of the gradients is used. The equations for the interior points are identical to those obtained with a traditional finite-difference approach, however the handling of the surface condition (8.44) is facilitated by the finite-volume representation. The discrete values of the temperature are taken at the cell centers, except for the surface temperature  $T_s$ . Each slab has a uniform mesh, with cell sizes  $\Delta x_-$  and  $\Delta x_+$ . The domain for  $x < 0$  has  $N_-$  cells, and the domain  $x > 0$  has  $N_+$  cells. The space discretisation is sketched in Figure 8.1.

### Interior points

Each cell from the interior of a domain is subject to the following ODE:

$$d_t T_i = \frac{D_{\pm}}{\Delta x_{\pm}^2} (T_{i+1} - 2T_i + T_{i-1}) \quad (8.46)$$

with the thermal diffusivity  $D_{\pm} = \lambda_{\pm}/(\rho_{\pm}c_{\pm})$ .

For the point  $i = -1$  this ODE is replaced by:

$$\Delta x_- \rho_- c_- d_t T_{-1} = -\lambda_- \frac{T_{-1} - T_{-2}}{\Delta x_-} + \lambda_- \frac{T_s - T_{-1}}{\frac{1}{2}\Delta x_-} \quad (8.47)$$

which can be reformulated as:

$$d_t T_{-1} = \frac{D_-}{\Delta x_-^2} (T_{-2} + 2T_s - 3T_{-1}) \quad (8.48)$$

Similarly, we obtain for the other side:

$$d_t T_1 = \frac{D_+}{\Delta x_+^2} (T_2 + 2T_s - 3T_1) \quad (8.49)$$

### Surface connection conditions

The interface connection condition (8.44) is discretised with a first-order approximation of the gradients:

$$\lambda_- \frac{T_s - T_{-1}}{\frac{1}{2}\Delta x_-} = \lambda_+ \frac{T_1 - T_s}{\frac{1}{2}\Delta x_+} + \Phi_s(T_s) \quad (8.50)$$

Analogously to Chapter 4, the surface temperature is an algebraic variable of index 1. When the values of  $T_1$  and  $T_{-1}$  are provided,  $T_s$  can be found by solving the previous equation, which is a nonlinear problem (because of  $\Phi_s$ ) of the form:

$$0 = \Psi_s(T_s, T_{-1}, T_1) \quad (8.51)$$

This problem can be solved via fixed-point iterations, or via a Newton method. In all cases, convergence is set to a relative error of  $10^{-12}$ , so that errors on  $T_s$  do not affect the precision of the overall solution for the complete problem. The derivative  $\partial_{T_s} \Psi_s$  is never zero in practical cases. Consequently, the implicit function theorem states the existence of a continuous function  $h$  such that:

$$T_s = h(T_{-1}, T_1) \quad (8.52)$$

### 8.2.3 Imposing the surface connection condition

We can replicate the situation where 2 solvers are coupled and must communicate at discrete coupling times to exchange data. Thus, we attribute each domain to a separate heat diffusion solver. The question now arises: how should the surface connection condition (8.50) be imposed? At least three boundary conditions can then be applied to the separate solvers:

**Dirichlet boundary condition** The coupling algorithm imposes the value of  $T_s$ , from which, via Equations (8.48) and (8.49), the evolutions of the cells surrounding the interface can be computed.

**Neumann boundary condition** In that case, the coupling algorithm imposes the heat flux that enters each slab, thus Equations (8.48) and (8.49) are conceptually replaced by:

$$d_t T_{-1} = \frac{1}{\Delta x_- \rho_- c_-} \left( -\lambda_- \frac{T_{-1} - T_{-2}}{\Delta x_-} + \Phi_- \right) \quad (8.53)$$

$$d_t T_1 = \frac{1}{\Delta x_+ \rho_+ c_+} \left( \lambda_+ \frac{T_2 - T_1}{\Delta x_+} - \Phi_+ \right) \quad (8.54)$$

where:

$$\Phi_- = (\lambda_- \partial_x T)(0^-) = \lambda_- \frac{T_s - T_{-1}}{\frac{1}{2} \Delta x_-} \quad (8.55)$$

$$\Phi_+ = (\lambda_+ \partial_x T)(0^+) = \lambda_+ \frac{T_1 - T_s}{\frac{1}{2} \Delta x_+} \quad (8.56)$$

which are computed by the coupling algorithm once  $T_s$  has been determined.

Technically, ODEs (8.53) and (8.54) are not used, but rather the original ODEs (8.48) and (8.49), where  $T_s$  is computed with a ghost-point approach:

$$T_{s,\pm} = T_{\pm 1} \mp \frac{1}{2} \Delta x_{\pm} \frac{\Phi_{\pm}}{\lambda_{\pm}} \quad (8.57)$$

Note that these ghost-point values may be different for each slab.

**Balance condition** The previous cases are adapted to the situation where both heat solvers are not able to account for the surface connection condition. Thus, this condition is solved by the coupling algorithm to compute the correct value of  $T_s$ , which then imposes a Dirichlet or Neumann boundary condition to each solver.

Another possibility is to let one or both solvers have the ability to determine  $T_s$  on their own, by solving the nonlinear problem (8.51). In that case the corresponding solver is given the value of the temperature in the cell nearest to the surface in the other domain. For instance, the solver for  $T_-$  ( $x \in [-L_1, 0]$ ) is given the value of  $T_1$ ,  $\Delta x_+$  and  $\lambda_+$ , so that it can solve Equation (8.51) on its own. Each solver thus computes its surface temperature  $T_{s,\pm}$ , which is imposed as a Dirichlet boundary condition. This temperature may be different in both solvers.

This balance condition can be seen as a shifted Dirichlet condition, where the prescribed temperature is  $T_{\pm 1}$  instead of  $T_s$ . The added complexity is that of solving the surface condition (8.50) in one solver. Typically, the other solver can provide a handle to a function which, depending on  $T_s$ , returns the corresponding heat flux on the other side. Note that using this balance condition for one of the slab solvers is equivalent to introducing  $T_s$  as a discrete variable, which is algebraic of index 1. In the coupling between CHARME and VULC1D from Chapter 7, the solid propellant solver uses a balance condition, while the fluid solver uses a Neumann condition.

## 8.2.4 Matrix form of the equations

Let us write the complete equations. To better distinguish the role of the various interface condition formulations, we separate their contributions from the inner point dynamics. We consider the state vector  $X = (T_{-N_-}, \dots, T_{-1}, T_1, \dots, T_{N_+})^t$ . The complete system reads:

$$M \frac{dX}{dt} = AX + b(X) \quad (8.58)$$

with:

$$M = \text{diag}\left(\frac{\Delta x_-^2}{D_-}, \dots, \frac{\Delta x_-^2}{D_-}, \frac{\Delta x_+^2}{D_+}, \dots, \frac{\Delta x_+^2}{D_+}\right) \quad (8.59)$$

$$A = \left( \begin{array}{ccc|ccc} -1 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -3 \\ \hline & & & -3 & 1 & \\ & & & 1 & -2 & 1 \\ & & & & \ddots & \ddots & \ddots \\ & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -1 \end{array} \right) \quad (8.60)$$

$$b(X) = (0, \dots, 0, 2h(T_{-1}, T_1), 2h(T_{-1}, T_1), 0, \dots, 0)^t \quad (8.61)$$

The term  $Ax$  represents the independent evolution of each slab. The term  $b(X)$  represents the coupling terms, which are nonlinear if a nonlinear surface reaction term is included, as in Equation (8.50). In the monolithic case, both slabs are advanced as a single system, hence the flux  $\Phi_{\pm}$  are themselves functions of  $T_{\pm 1}$ , hence the simplified form of the function  $h$  which constructs  $T_s$ .

A rapid verification of the system can be performed by considering the overall rate of increase for the thermal energy  $E_{\pm}$  in one slab, for instance the one at  $x < 0$ :

$$d_t E_- = \sum_{i=-N_-}^{-1} \rho_- c_- \Delta x_- d_t T_i = \dots = \frac{\Delta x_- \rho_- c_- D_-}{\Delta x_-^2} (2T_s - T_{-1}) = \lambda_- \frac{T_s - T_{-1}}{\frac{1}{2} \Delta x_-} \quad (8.62)$$

We recover the first-order approximation of the heat flux  $(\lambda \partial_x T)(0^-)$ , thus the energy balance is coherent. For the whole system, we also obtain the expected energy variation:

$$d_t (E_- + E_+) = \Phi_s(T_s) \quad (8.63)$$

## 8.3 Co-simulation framework

For the co-simulation, we split the state vector into  $X_-$  and  $X_+$ , representing the discrete values of the temperature field in each slab.

During a coupling time step, each slab evolves independently, governed by the following equations:

$$M_{\pm} \frac{dX_{\pm}}{dt} = A_{\pm} X_{\pm} + b_{\pm}(T_{\pm 1}, \hat{T}_{\mp 1}, \hat{T}_s, \hat{\phi}_{\pm}) \quad (8.64)$$

Here the notation  $\hat{\cdot}$  indicates that the evolution of the other slab, which is required to compute the coupling terms, is an assumed one. This is typically the polynomial extrapolation defined by Equation (8.21) based on the values at the previous couplings steps. The term  $b$  can take multiple forms, depending on the imposed boundary conditions, which we describe next.

### 8.3.1 Transmission condition

The difficulty in simulating the coupled evolution with separate slab solver lies in the exchange of fluxes through the interface condition (8.50), which is also referred to as transmission condition in the waveform relaxation community.

### Dirichlet boundary condition

In this configuration, the surface temperature  $T_s$  that satisfies the surface connection condition (8.50) is computed at each coupling step by the coupling algorithm and its prediction  $\hat{T}_s$  is imposed at the boundary  $x = 0$  of each slab for the next coupling step.

In that case, the term  $b$  is simply:

$$b(t) = \underbrace{(0, \dots, 0, 2\hat{T}_s(t))}_{b_-(\hat{T}_s(t))}, \underbrace{(2\hat{T}_s(t), 0, \dots, 0)}_{b_+(\hat{T}_s(t))} \quad (8.65)$$

A time dependence is therefore introduced following the use of the polynomial predictor.

### Neumann boundary condition

In that case as well, the surface temperature  $T_s$  that satisfies the surface connection condition (8.50) is computed at each coupling step by the coupling algorithm and the heat fluxes  $\Phi_{\pm} = (\lambda_{\pm} \partial_x T)$  ( $0^{\pm}$ ) are predicted as  $\hat{\Phi}_{\pm}$  and imposed at the boundary  $x = 0$  of each slab for the next coupling step.

In that case, the term  $b$  reads:

$$b(X_-, X_+, t) = \underbrace{(0, \dots, 0, 2T_{-1}(t) + \frac{\Delta x_-}{\lambda_-} \hat{\Phi}_-(t))}_{b_-(X_-, \hat{\Phi}_-(t))}, \underbrace{(2T_1(t) - \frac{\Delta x_+}{\lambda_+} \hat{\Phi}_+(t), 0, \dots, 0)}_{b_+(X_+, \hat{\Phi}_+(t))} \quad (8.66)$$

### Balance boundary condition

Here, each solver dynamically computes the surface temperatures that satisfies Equation (8.50). However this equation requires the knowledge of  $T_1$  and  $T_{-1}$ . To enable a split integration, one of these values must be predicted. Thus, the coupling algorithm constructs a prediction  $\hat{T}_1$  (resp.  $\hat{T}_{-1}$ ) and sends it to the solver for  $x < 0$  (resp.  $x > 0$ ). Then, each solver can compute its own reconstructed surface temperature, in the same spirit as Equation (8.52):

$$T_{s,-}(t, T_{-1}) = h(T_{-1}, \hat{T}_1(t)) \quad (8.67)$$

$$T_{s,+}(t, T_1) = h(\hat{T}_{-1}(t), T_1) \quad (8.68)$$

Consequently, the term  $b$  reads:

$$b(X_-, X_+, t) = \underbrace{(0, \dots, 0, 2h(T_{-1}, \hat{T}_1(t)))}_{b_-(X_-, \hat{T}_1(t))}, \underbrace{(2h(\hat{T}_{-1}(t), T_1), 0, \dots, 0)}_{b_+(X_+, \hat{T}_{-1}(t))} \quad (8.69)$$

It is known that Dirichlet boundary conditions are the most stable ones for conjugate heat transfer problems. Indeed, in such configurations, the limiting factor is the lack of thermal inertia for the gas phase compared to the solid, which in our case may be mimicked by specifying a much smaller density for the right slab. Imposing a constant Dirichlet condition to that slab will enforce a certain stability of its near-surface temperature profile. On the opposite, imposing a constant (say positive) heat flux via a Neumann condition can lead to a fast temperature increase. If the coupling time step is small enough, this increase is sufficiently well contained and simply causes the imposed heat flux to diminish for the next time step. However, if the time step is too large,  $T_1$  will rise too quickly so that it will become larger than  $T_s$ . Thus, the heat flux becomes negative at the next

coupling time step, and an oscillating instability of the heat flux develops, which usually leads to a divergence of the simulation. Therefore, it is common to associate a Dirichlet boundary condition with the fastest reacting system, while a Neumann condition is well suited to the slowest one [256].

### 8.3.2 Coupling procedure

We now describe how the coupling is performed for one coupling time step. The explicit and implicit procedures from Section 8.1.1 have been implemented. The key points of the procedure are given in Algorithm 1. At the beginning of a coupling step from time  $t_n$  to  $t_{n+1}$ , the surface conditions are solved to compute the initial values of the surface variables and fluxes at time  $t_n$ . The predictors, which are wrappers around a Newton interpolation formula, receive these values, allowing for an extrapolation on the time interval  $[t_n, t_{n+1}]$  to be computed.

The first WR iteration is then performed. A Jacobi update of the subsystems is conducted, using these extrapolations. Each subsystem is integrated with the adaptive ODE solver LSODA, a Python wrapper around a Fortran multistep integrator [276, 277], using a relative integration error tolerance of  $10^{-13}$ , so that the error is negligible compared to the error induced by the polynomial approximations of the inputs. The master algorithm can then compute the surface variables at  $t_{n+1}$ . The predictor polynomials are updated with these new values, so that an interpolation of the surface variables on  $[t_n, t_{n+1}]$  can be provided for the next iteration. Note that, to keep the same number of sampling points for the Newton formula (constant order), the oldest sampling point is discarded.

This whole WR iteration is performed anew, with the updated predictors. The iterative process stops once the relative variation of the surface variables from one iteration to the next is lower than a user-specified bound. Typically, we use the relative and absolute tolerances  $rtol = atol = 10^{-11}$ . The convergence of the temperature profiles in the slabs can also be considered, however we found that monitoring that of the surface temperature is sufficient. Once convergence is attained, the state vector at time  $t_{n+1}$  is returned. The convergence of the WR iteration is linear in this case, since it is a simple fixed-point iteration. Acceleration methods that enable a higher-order convergence rate have not been considered in this initial work.

## 8.4 Assessment of the convergence

The previous model and co-simulation structure have been implemented in Python. Some details of the implementation are given in Chapter 10. The chosen properties are  $\lambda_{\pm} = 1$ ,  $c_{\pm}=1$ ,  $\rho_- = 50$  and  $\rho_+ = 1$ . The ratio of each property between both slabs is comparable to that found at the surface of a burning solid propellant. For simplicity, we only consider dimensionless temperatures which are on the order of 1. The surface reaction has the following parameters:  $T_a = 1$ ,  $A = 100 \text{ W.m}^{-2}$ , and  $T_b = 1$ . The initial temperature field is uniform at  $T = 0.2$ . We set  $L_1 = L_2 = 1 \text{ m}$ , and each slab is discretised with 19 cells.

### 8.4.1 Dynamics of the toy model

A reference solution is obtained with a monolithic approach, using the ODE solver LSODA and a relative error tolerance of  $10^{-13}$ . The corresponding evolution of the temperature field and surface temperature are shown in Figure 8.2. The surface reaction releases heat which leads to an increase of the surface temperature. The right slab, whose density is lower than that of the left one, heats

---

**Algorithm 1** Single co-simulation step
 

---

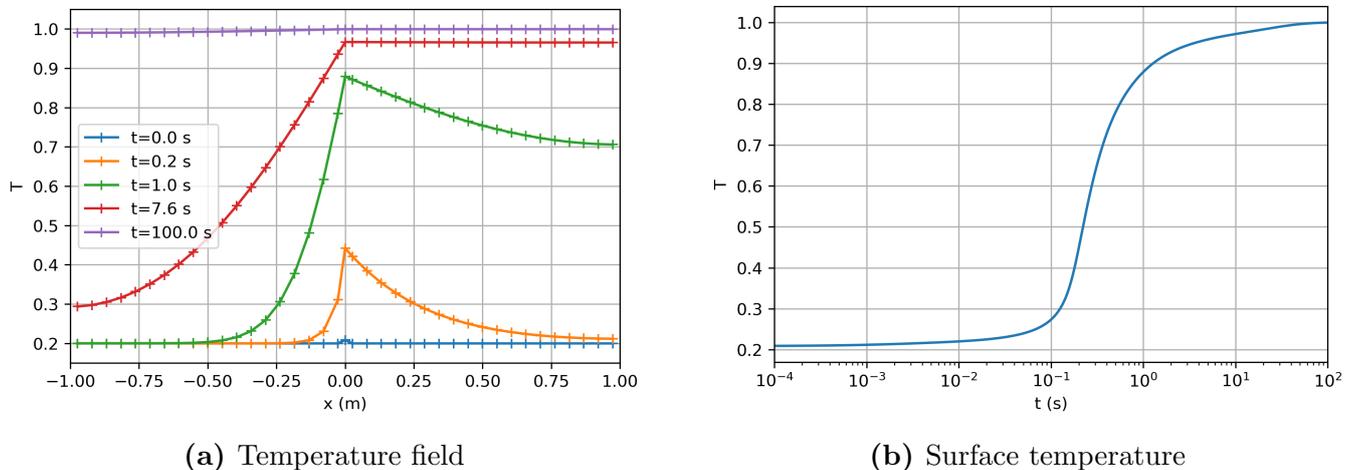
```

1: procedure COUPLINGSTEP( $t_n, X_n, \Delta t$ )
2:   Inputs:  $t_n, X_n, \Delta t$  starting time  $t_n$ , complete state vector at  $t_n$ , time step
3:   Outputs:  $X_{n+1}$ , ierr solution at time  $t_{n+1}$ , error code
4:
5:    $t_{n+1} = t_n + \Delta t$ 
6:    $T_{s,n}, \Phi_{\pm,n} = \text{computeSurface}(X_n)$  solve surface condition at  $t_n$ 
7:    $\text{updatePredictors}(t_n, T_{s,n}, T_{\pm 1}, \Phi_{\pm})$  update polynomials
8:    $k=0$ 
9:   while True do WR loop
10:     $k = k + 1$ 
11:     $T_{s,pred} = \hat{T}_s(t_{n+1})$ 
12:     $X_{-,n+1}^k = \text{integrateSubsystem1}(X_{-,n}, t_n, \Delta t, \text{predictors})$ 
13:     $X_{+,n+1}^k = \text{integrateSubsystem2}(X_{+,n}, t_n, \Delta t, \text{predictors})$ 
14:     $T_{s,n+1}^k, \Phi_{\pm,n+1}^k = \text{computeSurface}(X_{n+1}^k)$  update surface variables
15:     $\epsilon_k = \left| \frac{T_{s,pred} - T_{s,n+1}^k}{\text{atol} + \text{rtol} |T_{s,n+1}^k|} \right|$  error on  $T_s$ 
16:     $\text{updatePredictors}(t_{n+1}, T_{s,n+1}^k, T_{\pm,n+1}^k, \Phi_{\pm,n+1}^k)$ 
17:    if  $\epsilon_k < 1$  then WR has converged
18:      ierr=0
19:       $X_{n+1} = X_{n+1}^k$ 
20:      return
21:    end if
22:     $q = \frac{\epsilon_{k-1}}{\epsilon_k}$  compute WR convergence rate
23:     $k_{req} = k + \frac{\log(\epsilon_k)}{\log q}$  predicted number of iterations
24:    if  $k > k_{max}$  or  $k_{req} > k_{max}$  or  $q < 1$  then poor convergence or divergence
25:      ierr=1
26:       $X_{n+1} = X_{n+1}^k$ 
27:      return
28:    end if
29:  end while
30: end procedure

```

---

up quickly and heat diffuses faster to its outer boundary. The left slab has more thermal inertia, thus heats up more slowly. The non-equality of the heat fluxes on both sides of the surface is clearly apparent and is a measure of the surface heat release  $\phi_s$ .



**Figure 8.2** Dynamics of the toy model

### 8.4.2 Co-simulation results

We now wish to investigate how the co-simulation behaves for that particular problem. For that purpose, we focus on a time period of 0.5 s, starting from time  $t_0$  for which the reference solution reaches  $T_s = 0.5$ . We then perform multiple co-simulations with different numbers of time steps and several choices of boundary conditions. To ensure that the polynomial predictors reach their full order directly at the first co-simulation step, they are initialised with sampling points taken from the reference solution for  $t \leq t_0$ .

Three sets of surface boundary conditions are tested: Dirichlet-Dirichlet (requiring a predictor  $\hat{T}_s$ ), Neumann-Balance (requiring  $\hat{\Phi}_-$  and  $\hat{T}_{-1}$ ), and Balance-Neumann (requiring  $\hat{\Phi}_+$  and  $\hat{T}_1$ ). The latter is representative of the coupling between VULC1D and CHARME.

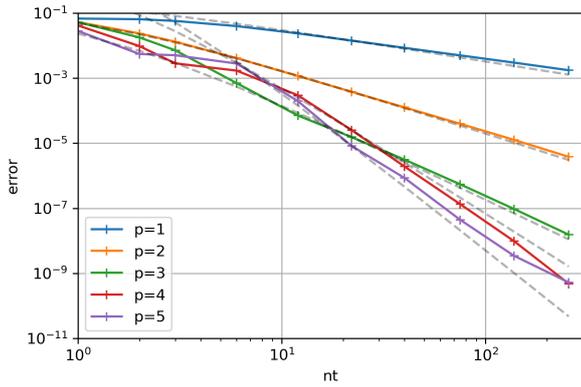
In Figure 8.3, we show how the co-simulation results converge towards the reference simulation. The error monitored is a global relative error on the temperature fields in both slabs, in time and space, using the  $L_1$ -norm:

$$\epsilon = \frac{1}{L_1 + L_2} \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \int_{-L_1}^{L_2} \left| \frac{T(x, t) - T_{ref}(x, t)}{T_{ref}(x, t)} \right| dx dt \quad (8.70)$$

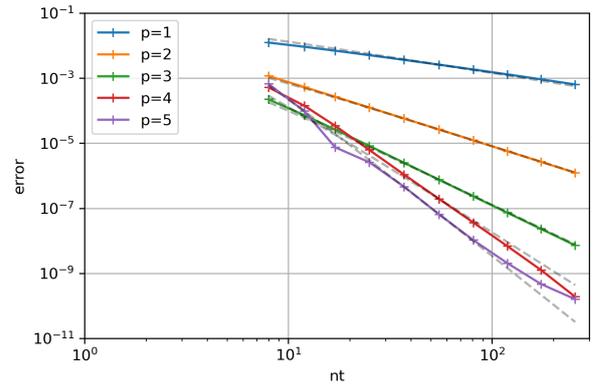
The integrals are evaluated numerically using the simple rectangle formula, with the reference solution  $T_{ref}$  being interpolated on the same time grid as the selected co-simulation solution  $T$ .

We clearly see that, in the asymptotic regime, the order of convergence is equal to the degree of the prediction polynomials. The convergence curves are only plotted when a stable solution has been found. In particular, the Neumann-Balance and Balance-Neumann configuration diverge during the WR iterations when the coupling time step is too large, whereas the Dirichlet-Dirichlet condition is more stable.

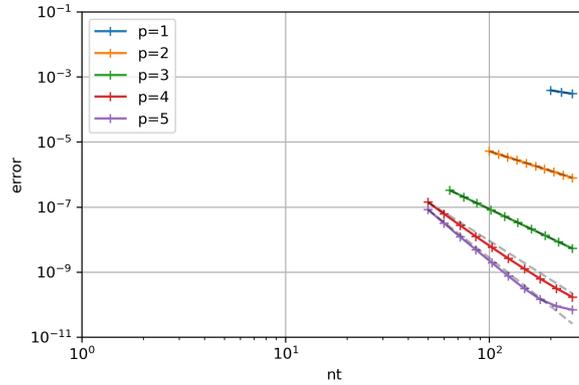
An interesting observation is that the absolute error levels are slightly different between each configuration. Taking the convergence curves for third degree polynomials, we see that the Dirichlet-



(a) Dirichlet-Dirichlet



(b) Neumann-Balance



(c) Balance-Neumann

**Figure 8.3** Convergence of the global error for different prediction orders  $p$ . Grey dashed curves corresponds to the theoretical orders.

Dirichlet setup has an error level approximately twice that of the Neumann-Balance configuration, which itself roughly twice that of the Balance-Neumann condition.

However, we also see that the stability of the coupling is vastly different. Indeed, the largest stable time step for the Dirichlet-Dirichlet configuration is 8 times larger than with the Neumann-Balance one, and approximately 50 to 100 times larger than with the Balance-Neumann condition. Hence, we clearly see that the transmission conditions are of particular importance for the accuracy and stability of the coupling. Following the various sources of numerical errors (subsystem integration, WR convergence criterion...), the relative error cannot reach levels below  $10^{-11}$ .

Overall, this study with the toy model shows that the co-simulation approach indeed enables to reach high orders of convergence. We now turn to another test case which has been investigated during this exploratory work to study the ability to dynamically adapt the coupling time step.

**Remark 14.** *The coupling between VULC1D and CHARME from Chapter 7 uses a Balance-Neumann condition, however the time step levels are such that stability issues do not arise. While it could be interesting to investigate the use of other boundary conditions in that case, the previous choice is mainly dictated by the fact that CHARME cannot solve the surface connection condition itself, whereas VULC1D natively possesses this ability. Also, the Neumann condition applied to the CFD solver after the time advancement of VULC1D allows for an exact energy conservation to be ensured,*

since the imposed fluxes are those that have actually left VULC1D.

## 8.5 Time step and order adaptation

One of the initial aims of this chapter is to use the co-simulation framework to automatically adapt the coupling time step, following the approach discussed in Section 8.1.3. We now wish to investigate this aspect and demonstrate it on a simple case. To prepare for a future use with VULC1D and the CFD toolchain CEDRE, we propose to consider a simple system, representative of that coupling, while also involving VULC1D. It consists of zero-dimensional combustion chamber model, which is coupled with VULC1D to simulate the ignition of an SRM. The chamber is simply modelled with the equations presented in Section 1.2.3. The mass flux  $m$  from the propellant is the one produced by the 1D propellant model. The chamber temperature is assumed constant at 3500 K, so that an equation governing the evolution of its temperature is not necessary. This is also reminiscent of early 0D SRM ignition models [43]. This model is not able to provide a heat flux that can be applied to the propellant surface. Hence, to attain ignition, a constant heat flux is imposed at the propellant surface in VULC1D.

Contrary to the toy model, this coupling does not involve an exchange of heat fluxes localised at a physical interface. Only VULC1D provides fluxes (here of mass) to the chamber solver, whereas the latter imposes the chamber pressure as the uniform gas-phase pressure for VULC1D.

The one-dimensional code VULC1D is modified such that pressure can be prescribed as a polynomial of time. This polynomial is constructed using the previous co-simulation approach and is updated at each coupling step. A similar polynomial is constructed for the mass flow rate  $m$  injected by the propellant model inside the combustion chamber.

Physical realism is not the focus of this simple configuration. We only consider this test case as a simple benchmark that features the main dynamical aspects encountered in a complete model of a combustion chamber (pressure variation, ignition of the propellant...).

The chamber model and the co-simulation framework have been implemented in Fortran. A single iteration of the WR procedure is performed, so as to be closer to the technical restrictions for a future implementation in CHARME. To ensure that the main source of error is that of the polynomial extrapolation, the chamber model (a single scalar ODE on  $P$ ) is integrated with 1000 explicit Euler time steps per coupling step, while the 1D propellant model from VULC1D is integrated adaptively with ESDIRK-54A and an error tolerance of  $10^{-8}$ .

### 8.5.1 Dynamics of the 0D chamber ignition

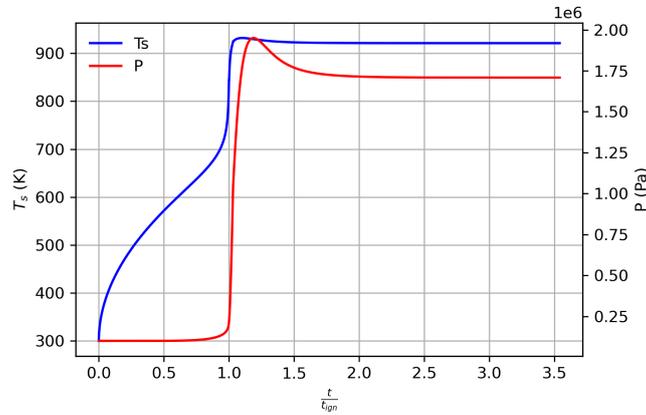
The case is representative of the 2D-axisymmetric fictitious motor TEP which will be discussed in more details in Chapter 11. The chamber is 17 cm long and has a radius of 4.5 cm. Accounting for the nozzle, the total volume is  $1.2 \times 10^{-3} \text{ m}^3$ . Initially, the internal pressure is set to  $P = 10^5 \text{ Pa}$  and the nozzle is sealed.

The parametrisation of the solid propellant model from VULC1D is a slight variation of that from Section 9.4, which yields a lower steady-state temperature. The solid phase is initially at  $T_0 = 300 \text{ K}$ . It is heated up from  $t = 0$  onwards by a laser providing a  $2 \text{ MW.m}^{-2}$  heat flux absorbed at the surface.

Once the solid propellant is ignited, the laser is shut down, and combustion gases coming from the 1D model accumulate in the chamber, causing a rise in pressure. The nozzle seal is removed when the chamber pressure becomes larger than  $P = 10^6 \text{ Pa}$ , and the nozzle throat is assumed

instantaneously choked. The mass flow rate exiting the combustion chamber can then be computed from Equation (1.8), where the outer pressure is assumed to be  $10^5$  Pa.

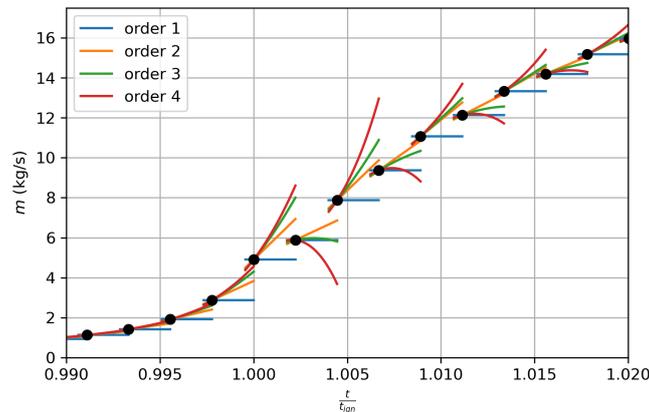
Figure 8.4 shows the coupled dynamics. We clearly see the initial inert heating, the onset of the pyrolysis gases flow inside the chamber, and the stabilisation onto steady-state.



**Figure 8.4** Coupled dynamics of the solid propellant and the combustion chamber.

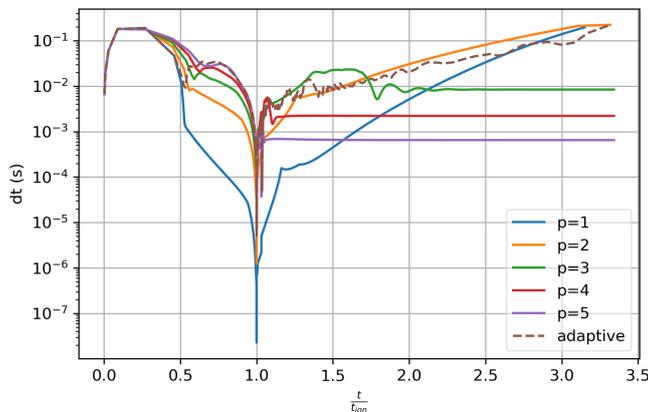
### 8.5.2 Effect of the prediction order

Let us investigate the effect of the prediction order on the coupling values. Figure 8.5 gives a visual representation of the polynomial predictions used for the co-simulation. We see that high-order predictions are more accurate when the dynamics is smooth. However, when the dynamics changes (here with the laser cut-off), the previous data points that are used for high-order extrapolation are not coherent any more with the new trajectory and cause important errors in the predicted values. This indicates that a variable order would be very interesting to improve the accuracy of the computation, increasing the order as much as possible during smooth phases, and reverting to low order when a change of dynamics is detected.



**Figure 8.5** Visualisation of the predictive polynomials for various orders, around the time of ignition (laser cut-off). The black dots represent the surface temperature computed by VULC1D at each coupling step.

We can also study the impact of the prediction order on the time step required to match a specific error tolerance on the coupling variables. Figure 8.6 shows the evolution of the time step for different orders when requiring a relative error tolerance of  $10^{-4}$  on the accuracy of the polynomial predictions for  $m$  and  $P$ . The orders for the extrapolations of  $m$  and  $P$  are identical. We see that during the highly transient phases, higher orders are able to use larger coupling time steps. On the opposite, when the coupled dynamics tends to steady-state, the time step is restricted to very low values with high-order predictions. This latter fact is easily explained: when the solution is stable and does not vary much between consecutive steps, the high-order terms have a tendency to diverge rapidly when extrapolating. Therefore, the prediction error becomes large and imposes a severe time step restriction on the coupling. Conversely, low-order polynomials are more stable for the extrapolation of nearly constant variables.



**Figure 8.6** Evolution of the time step for various prediction orders.

This experiment suggests that a variable order could lead to considerable gains in computational efficiency by minimizing the number of coupling steps. Consequently, in the following, we discuss how the prediction order can be dynamically adapted.

### 8.5.3 Order adaptation

Varying the order of the polynomial prediction is very similar to the order variation found in multistep methods (Adams and BDF) in the context of the solution of ODEs. For these multistep methods, two main approaches exist [218]:

- For each predicted variable, we choose the order that permits the largest step size. This is used for example in the LSODE method [278].
- We check the coherence of the Taylor expansion for each extrapolated variable  $u$ :

$$u(t + \Delta t) = \sum_{j=0}^k \frac{\Delta t^j}{j!} u^{(j)}(t) + \mathcal{O}(\Delta t^{k+1}) \quad (8.71)$$

This series is considered “well-behaved” when the norm of the successive terms decreases as  $j$  increases. For instance, if the current order is  $p$ , we observe the sequence  $|h^{p-1}u^{(p-1)}|$ ,  $|h^p u^{(p)}|$ ,  $|h^{p+1}u^{(p+1)}|$ ,  $|h^{p+2}u^{(p+2)}|$ . If this sequence is increasing, the order is lowered, otherwise the

order can be increased. Petzold [218] indicates that this approach allows for the early detection of instabilities arising from the polynomial extrapolation. This approach is used in the well-known DAE integrator DASSL [279]. Numerically, the successive temporal derivatives of  $u$  can be approximated with the divided difference coefficients [218]:

$$u^{(j)} \approx j! \delta^j u \quad (8.72)$$

which are readily accessible when using a Newton formula for the extrapolation.

To our knowledge, a single reference describes the use of dynamic order adaptation in the co-simulation literature [280], relying on the second method. In the present manuscript, we propose to use the first method instead. Let us expose the basic principles. After each coupling step, the errors in the predicted variables are assessed *a posteriori* for each possible order (typically 1 to 5), and the corresponding optimal coupling time steps (which lead to errors equal to the relative error tolerance) are computed based on Equation (5.41) and the order of the polynomial approximation. The order that yields the largest optimal time step is selected. This approach enables a quick lowering of the order when a rapid change in the temporal dynamics of the coupling variables occurs. To avoid repeated order changes, we only allow an order increase if 3 coupling steps have been successful with the current order.

Note that if advanced PI or PID time step controllers are used to smooth the evolution of the time step as in traditional ODE solvers, issues may arise upon varying the order. Indeed, these controllers usually assume a constant order of convergence for the local truncation error. If the order is dynamically changed, the history of truncation errors used by the controller is no longer coherent, because the error constant and its dependence on  $\Delta t$  is modified. A possible fix is to use an alternative PID formulation with accounts for order variation, e.g. the one presented in [281], which has been successfully applied to BDF methods. Here, we use the simpler time step controller from Section 5.1.5, which does not suffer from this issue.

The proposed order adaptation method has been implemented in the Fortran co-simulation tool. Figure 8.7 shows the order for the extrapolation of the parietal pressure  $P$ . Similar results are obtained for the other coupling variables. The order is gradually increased during the inert phase, as more and more sampling points become available. The evolution of the different variables is very smooth and enables a high-order resolution thanks to the high-order prediction. As the laser is cut-off, the order locally reverts to 2 before increasing back again to orders 3 or 4. Finally, as the solution becomes steady, the order is gradually decreased to 1, allowing for large time steps to be used and for an efficient stabilisation of the solution around steady-state.

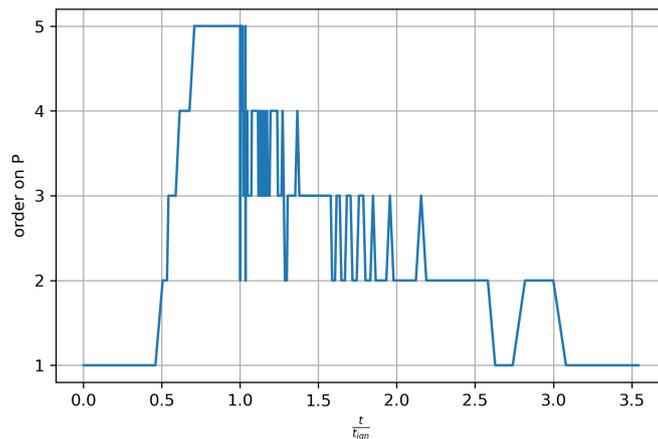
The time step history used by the adaptive order solution is plotted in Figure 8.6 (dashed brown line). We can observe that, at each step, the adapted coupling time step is very close to the maximum of the time steps taken by the fixed-order integrations, thus validating the behaviour of the order selection algorithm.

In this particular test with a relative error tolerance of  $10^{-4}$  on the predicted coupling variables, the number of time steps required are listed in Table 8.1.

order	1	2	3	4	5	adaptive
steps	55829	959	500	1194	3683	296

**Table 8.1** Number of time steps taken for each prediction order

We see that the first-order prediction (constant extrapolation) is very inefficient, as can be observed by the large time step drop during ignition in Figure 8.6. Increasing the order to 2 or



**Figure 8.7** Evolution of the prediction order

3 improves the situation largely. Increasing the order further results in an increase in the number of steps because of the previously mentioned issue of the quickly diverging extrapolation when the solution dynamics are nearing steady state. The adaptive order takes the best of all the possible prediction orders and greatly improves the computational efficiency. As with typical multistep methods for ODEs, it was also observed that lowering the error tolerance  $rtol$  leads to using smaller time steps and higher-order polynomials.

## Partial conclusion

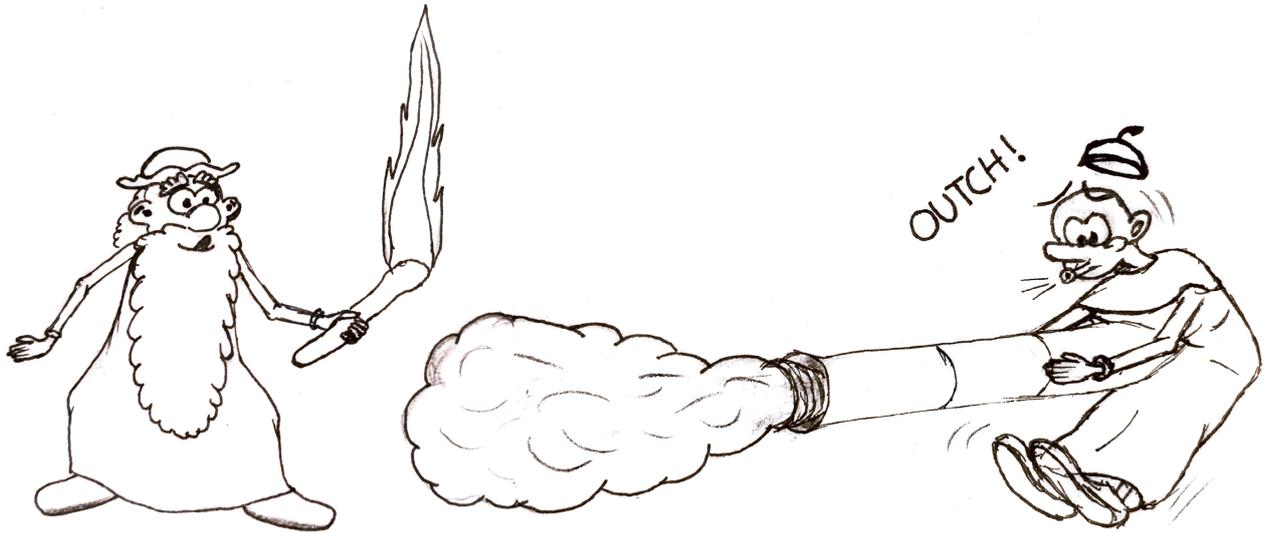
In this chapter, we have first envisioned several possibilities to increase the order of convergence in time of the coupling between VULC1D and CHARME. The idea of co-simulation, where the coupling variables and fluxes are extrapolated in time based on the previous coupling steps, appears to be well suited to the technical framework of CEDRE. A simplified theoretical analysis has shown that this approach indeed enables a high-order temporal resolution. A simple test case simulating the heat exchange between 2 slabs of different natures through a reactive surface has been designed and implemented in Python, and has demonstrated the high-order of convergence possible with the co-simulation approach.

With the aim of applying this method to the coupling from Chapter 7 between VULC1D and CHARME, we have implemented another case where VULC1D is coupled to a 0D SRM chamber. Dynamic adaptation of the coupling time step has been implemented and has suggested that low-order predictions should be used for slow dynamics, while higher-order predictions are more efficient for faster dynamics. Consequently, we have proposed a procedure to dynamically select the prediction order, which has proved reliable and efficient.

This chapter is a summary of early work that is still in progress. The exploration of the co-simulation approach for our applications should be continued. We believe that this technique should be able to yield high-order coupling with time adaptation. A particular point of interest, which has not been discussed in this chapter, is the issue of energy conservation that is capital in the conjugate heat transfer community for instance [241]. Co-simulation techniques usually do not conserve energy, however it is possible that modifications of the approach or on-the-fly flux corrections [242] could be useful to improve on that aspect.

# Part III

## Applications





In Part I, we have presented the development of a one-dimensional solid propellant ignition model and the associated numerical tool `VULC1D`. In Part II, we have discussed the coupling procedure between `VULC1D` and the CFD solver `CHARME` for the simulation of SRM ignition, in particular we described a first-order time accurate method, as a first step before the introduction of higher-order methods.

We introduced two coupled approaches:

- the 1D flame approach where the propellant flame is solved within `VULC1D` and appears as a boundary surface phenomenon to `CHARME`,
- the detailed approach where `VULC1D` only solves the solid phase evolution and the surface connection conditions, while `CHARME` is solving the entire propellant flame in the CFD domain itself.

The numerical and modelling frameworks are thus complete for the realisation of large-scale ignition simulations.

In this part, we propose to complete the path to such applied simulations. In particular, the models need to be parametrised. Useful characterisations of a solid propellant can be obtained with specific experiments. Therefore, in Chapter 9, we explain which experiments are available at ONERA, and how they enable a characterisation of a given propellant composition. Some model parameters can be directly obtained from such experiments, however many parameters cannot be explicitly deduced. Therefore, a constrained optimisation approach is developed to iterate on these remaining values.

In Chapter 10, we take time to underline the large implementation effort that has supported the work presented in this manuscript, e.g. development of `VULC1D`, as well as a series of dedicated tools, and give some insight on the technical choices made.

Finally, Chapter 11 presents the result of two-dimensional simulations of the ignition in a fictitious SRM combustion chamber. The new test case is designed such that all physical scales involved in the propellant and fluid models can be accurately resolved. Both coupled approaches from Chapter 7 are compared, and it is shown that the 1D flame approach gives very good results, nearly identical to those obtained with the detailed approach. This shows the capability of the developed simplified approach for the simulation of SRM ignition. This last chapter is the convergence point of all the efforts presented in this manuscript, truly showing the adequacy of the 1D flame approach for the fulfilment of the objectives of this thesis.



# Chapter 9

## Model fitting

### Summary

We present the experimental framework available at ONERA for the characterisation of solid propellants. We explain how their results can be used to determine the parameters of our 1D model. Some parameters can however not be simply related to measurements. Therefore a nonlinear optimisation procedure is used to generate a complete numerical configuration that matches experimental data.

In Part I we have presented the one-dimensional model and its numerical implementation. A first parametrisation has been presented in Section 3.3.3, utilising a simple kinetic scheme with two global species and a single irreversible reaction. This parametrisation is representative of the steady-state combustion of a specific AP-HTPB-Al solid propellant used for research purposes at ONERA [12]. In the present chapter, we investigate how the 1D model can be parametrised to reproduce the ignition dynamics of other solid propellants.

Most widely spread ignition models require that the thermal properties of the solid propellant be known, so as to compute its transient heating induced by the wall heat flux, and its steady-state characteristics (flame temperature, Vieille law coefficients...) for the description of quasi-steady combustion after ignition. The transition to ignition is determined by the attainment of an ignition temperature, which is traditionally fixed arbitrarily to physically-sound values [50, 53, 56, 58], or to values obtained via the fitting of a simple inert heating model to experimental laser ignition times [282]. It has however been shown that the actual ignition temperature may greatly depend on the heat flux level [60].

Slightly more advanced models developed at ONERA [52, 153] do not rely on a temperature criterion. In such models, ignition dynamically occurs by thermal runaway of surface or gas phase reactions, and the propellant flame is solved analytically with several assumptions, such as very large activation energies for the surface pyrolysis and gas phase reaction. From experimental measurements, dependence of the time to ignition (ignition delay) with respect to the surface heat flux can be obtained. To match this dependence, the few model parameters related to surface and flame processes (activation energies mainly) can be analytically fitted [27]. Alternatively, a least-square fitting process was also developed, where the parameters are iterated to maximise the similarity between the experimental and simulated evolution of  $T_s$  during ignition tests [283].

In the case of the generic unsteady model from Chapter 2, the number of parameters can be arbitrarily large, e.g. diffusion coefficients, number of species considered, reaction mechanisms...

Fitting our model can only be done if simplifying assumptions are used. We will therefore use the configuration from Section 3.3.3 as a baseline, keeping the simple gas phase representation (2 species, 1 reaction). Departing from this configuration by changing the values of the various parameters, it is hoped that various propellants can be modelled with acceptable accuracy.

In the present chapter, we will develop an optimisation procedure to automatically determine model parameters, so as to reproduce the behaviour of an AP-HTPB-Al research propellant characterised experimentally at ONERA. Fitting such a model to a real propellant requires a careful characterisation of the latter. For that purpose, we begin this chapter with an overview of the experimental framework from ONERA for the study of steady burning and ignition dynamics of a propellant sample, showing the various information that can be extracted from such experiments. In a second part, we explain how the corresponding experimental results can be linked to our model parameters, and we present an optimisation framework enabling the automatic determination of the previous model parameters to reproduce experimental measurements. Finally, optimisation results are presented and discussed.

## 9.1 Experimental data available at ONERA

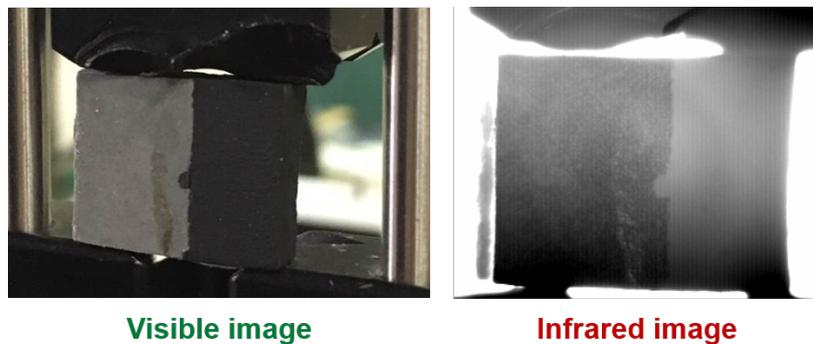
A large number of experimental apparatus have been used in the literature for the characterisation of solid propellants. Among these, the most important ones are those that enable the determination of the thermal properties of the solid phase, and those that focus on the determination of the steady-state regression rate and its dependence on pressure for instance. For the present manuscript, experiments related to the ignition dynamics of the propellant are also of great interest. In the following section, we describe some of the main experimental apparatus used at ONERA for these aspects.

### 9.1.1 Thermal and radiative measurements

ONERA has the ability to perform transient thermal experiments to determine the thermal diffusivity of a propellant sample via the use of the flash method [284, 285] with a diffusimeter. The basic principle is the following: a propellant sample is laid in a closed volume and its surfaces are insulated. A short-duration pulse of a high-power lamp heats up one of the surface of the sample. Temperature sensors monitor the temperature of that side and of the opposite one. Analysis of the temperature discrepancy between these two faces and its temporal evolution can be used to evaluate an effective thermal diffusivity using analytical solutions of the heat equation, regardless of the heterogeneity of the material. The heat capacity can also be evaluated by monitoring the temperature increase of a sample exposed to a predetermined energy income. The density is trivially measured by mass and volume considerations. Overall the thermal properties  $\lambda_c$ ,  $c_c$ ,  $\rho_c$  can be determined as functions of the temperature, or at least in an average manner, typically up to  $T \approx 500$  K. Note that the evolution of these properties at higher temperature levels is poorly characterised, because chemical reactions become important and render precise measurements difficult, and the degradation of the sample studied means that the properties measured are not those of the original sample any more.

For ignition experiments presented further, it is useful to monitor the propellant surface temperature during the initial heating phase. To avoid the insertion of thermocouples in the sample itself, an efficient alternative is the analysis of the infrared light emitted by the surface. Based on a simple thermal radiation model, the surface temperature can be determined. The knowledge of

the surface emissivity is however required. It is measured with the following process. A propellant sample is laid in an environment at constant temperature. Half of the sample's upper surface is painted with Nextel Velvet Coating, a black paint sold by Mankiewicz. The propellant sample is heated up to 50-60 °C and imaged by an infrared (IR) camera. Examples of the visible and infrared images obtained are shown in Figure 9.1. The surface emissivity of the paint is constant in the IR camera spectral range of interest (2.5 - 5.5 μm) within a 0.5% tolerance, with an average value of 0.97 [286]. The unpainted half of the sample displays a lower IR signal level than the Velvet-coated one, because its emissivity is markedly lower. The propellant surface emissivity is then evaluated with good error levels by comparing the IR signal levels of both halves. Values between 0.7 and 0.9 are typically obtained, depending on the propellant composition, the additives (e.g. carbon black), and surface state (e.g. rough, polished). Once the emissivity is known, the analysis of the IR signal intensity from each pixel of the IR camera footage enables the reconstruction of the surface temperature evolution in space and time during the exposition of the sample to the laser heat flux. Note that the measurements are not accurate any more once the propellant ignites, because the flame is not transparent to the IR radiation from the surface, and emits itself a much stronger IR light.



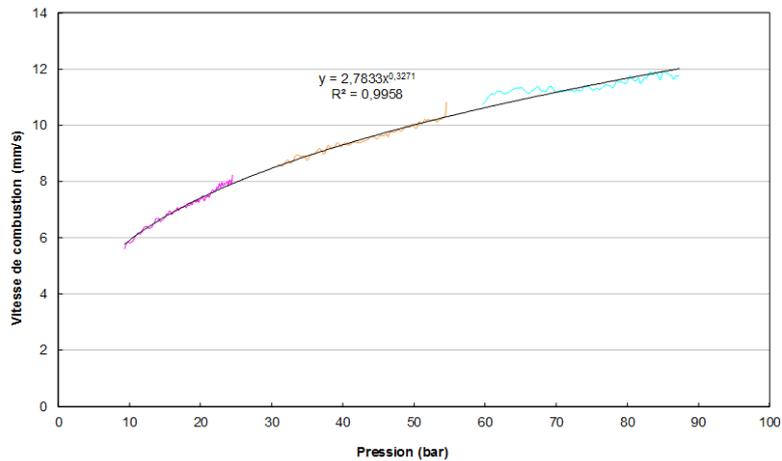
**Figure 9.1** Measurement of the propellant surface emissivity by comparison with a reference paint

### 9.1.2 Closed bomb

Another characterisation of paramount importance is that of steady-state burning. In particular, the coefficients of the Vieille law  $r = aP^n$  must be determined for the pressure range of interest, as constant or pressure-dependent variables. For that purpose, a widely used experiment, also performed at ONERA, is that of closed-bomb burning.

A propellant sample is placed within a closed volume (bomb) and ignited. The pressure evolution is monitored during the firing, and the surface regression rate is measured via the use of ultrasonic sensors. These emit sound waves at the bottom of the propellant, which are reflected on the propellant surface. Measurements of the delay between the emission of the signal and the reception of the reflected one lead to an estimation of the propellant thickness. Corrections need to be applied to account for the dependence on temperature and pressure of the ultrasonic wave propagation speed within the propellant [287]. Accuracy can be improved with additional refinements [288].

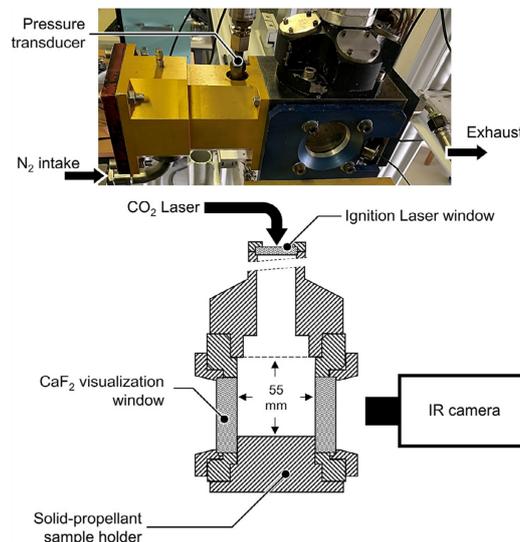
Performing the same test at various initial pressures enables the construction of the burning rate law. The classical Vieille law can then be fitted, at least piecewise, to the obtained data. Figure 9.2 shows the evolution of the measured pressure and regression rates for 3 tests, and the corresponding fit of the Vieille law for an AP-HTPB propellant from ONERA.



**Figure 9.2** Pressure and regression rates measured within a closed bomb (coloured lines), fitted Vieille law (black line)

### 9.1.3 Ignition dynamics

To study the ignition dynamics of a given propellant formulation, specific experiments need to be performed. In this section, we will only focus on the main test bench from ONERA, the CHAMADE test bench (Chambre d’Ablation des Matériaux Énergétiques) [25, 289]. This bench consists of a small pressurised chamber (a cube of 55 mm side length), and is presented in Figure 9.3. A small propellant sample (typically a cylinder with a diameter of 5 mm and a thickness of 2-3 mm) is placed at the bottom. The chamber can be pressurised with nitrogen up to 15 bars, but most tests are performed at 3 bars.



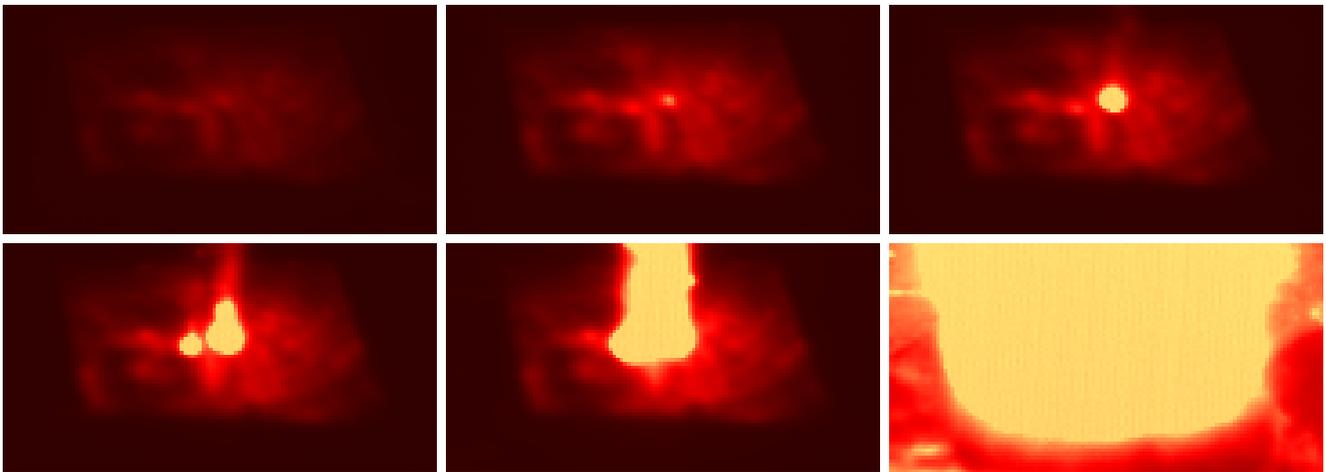
**Figure 9.3** Geometry and experimental setup of the CHAMADE test bench [25]

A CO<sub>2</sub> laser of wavelength 10.6 μm with a nominal power of 2 kW is used to heat up the sample by generating a radiative heat flux of known intensity (0.1 to 2 MW/m<sup>2</sup>) and duration (typically 2-200 ms). Infrared cameras are used to track the evolution of the propellant temperature on the whole surface before ignition, with an acquisition rate of 400-2000 Hz, depending on the camera

settings. Higher rates are used when the expected ignition delay is low. Pyrometers are also used to provide space-averaged temperature measurements at a higher rate of 2 kHz. Post-processing enables the construction of a representative surface temperature curve, and the determination of the ignition delay  $t_{igni}$ . That delay can be defined in various ways:

- time of appearance of the first light at the surface
- time of appearance of the first flame above the surface
- time when the representative surface temperature becomes 5% larger than the temperature achieved assuming an inert material, indicating an important development of the chemical reactions at and above the surface.

Example infrared images are presented in Figure 9.4, showing the initial state, the inert heating phase, the first ignition (also called “first light”), and the complete ignition of the sample. Example surface temperature measurements are shown in Figure 9.5 for different points at the surface.



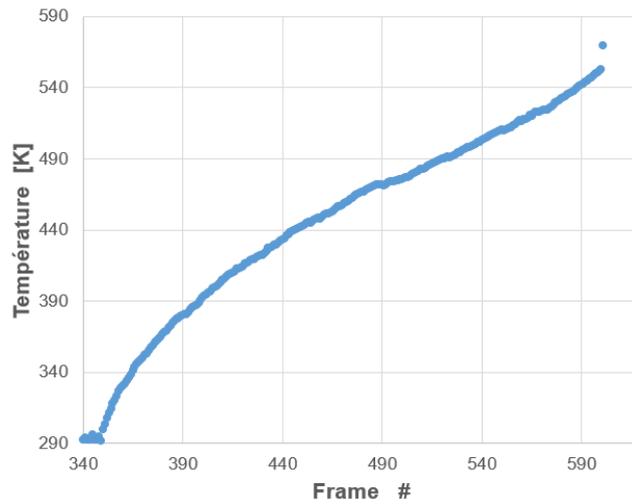
**Figure 9.4** Different phases of laser-induced ignition, as recorded by an infrared camera (courtesy of Robin Devillers, ONERA)

During the initial inert heating phase, the solid phase is fairly decoupled from the gas phase, such that the dynamics initially resembles that of an inert material without heat losses. If we consider that the whole incoming laser flux is constant and absorbed at the propellant surface, the surface temperature obeys the following law before ignition:

$$T_s(t) = T_i + \frac{q_{laser}}{\lambda_s} \sqrt{\frac{4\alpha}{\pi} t} \quad (9.1)$$

with  $q_{laser}$  the laser flux density, and  $\alpha = \frac{\lambda_s}{\rho_s c_s}$  the thermal diffusivity of the solid propellant. Since the thermal properties of the propellant are known from other laboratory experiments, the laser heat flux effectively absorbed can be computed by fitting the above formula on the initial part of the surface temperature history.

In the scope of these laser ignition tests, heat loss at the propellant surface via conduction to the gas phase can be neglected. Note that it could be accounted for by knowing the ratio of the effusivities of both phases, see [4].

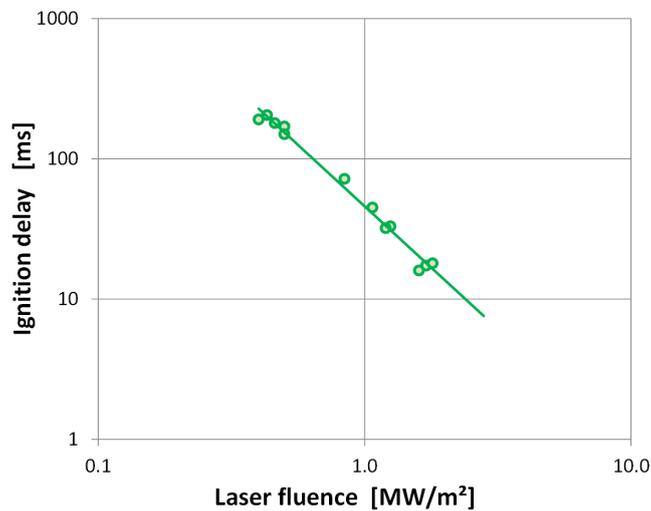


**Figure 9.5** Surface temperature history deduced from infrared images (courtesy of Robin Devillers, ONERA)

Multiple ignition tests can be performed with different laser power outputs, enabling the construction of the curve of ignition delay versus absorbed heat flux. A log-log fit of this curve will be referred to as the *ignition law* of the propellant:

$$t_{igni} = p_1 \log(q_{laser}) + p_0 \quad (9.2)$$

with  $p_1$  and  $p_0$  two constants which are fitted to the data. A typical ignition law is shown in Figure 9.6.



**Figure 9.6** Example ignition law obtained experimentally (courtesy of Robin Devillers, ONERA)

### 9.1.4 Other characterisations

In this chapter, only the results from the previously presented experiments will be used. However, for the sake of completeness, we briefly mention other less active experimental benches which could

allow for a more precise characterisation of a propellant to be obtained.

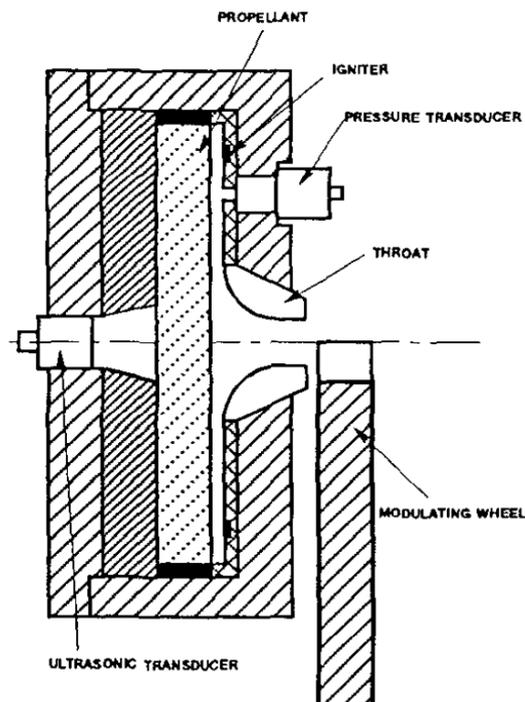
## Flame spread

The test bench CHEAP (CHambre d'Etude de l'Allumage des Propergols) enables the study of the self-sustained spreading of the propellant flame after ignition of a small portion of an elongated sample. This could be used to tailor the gas-phase heat release and the surface heat transfer in our model to reproduce the same flame propagation speed. It however requires at least a two-dimensional simulation, for example with the coupled approaches presented in Part II, which are costly and would not fit in the optimisation procedure presented further.

## Combustion response to pressure fluctuations

Another characterisation of interest is the non quasi-steady response of the pyrolysis mass flow rate  $m$  to fluctuations of the pressure, as a function of the frequency of these fluctuations. Various apparatus, e.g. T-burners [290] or laser-recoil benches [291], have been designed to measure this response, however data scatter is usually very large.

Another approach used at ONERA is that of a modulated exhaust jet burner. This approach relies on the use of a small combustion chamber where a propellant sample burns. The ONERA test bench is named LP8 and is described in [292]. A sketch of the bench is shown in Figure 9.7.



**Figure 9.7** Schematic view of the modulated exhaust test bench [292]

The chamber is linked to the outer atmosphere via a convergent nozzle, designed such that the exit nozzle is choked, allowing for the chamber to reach a relatively high level of pressure (15 bars). At the nozzle exit plane, a wheel with square teeth is rotated at high speed, with its teeth partially obstructing the exit surface of the nozzle. This produces a variation of the throat area, and consequently a chamber pressure variation which excites the propellant combustion. The oscillations

of the surface burning rate can be measured via an ultrasonic transducer. The analysis of the relative amplitude variations in regression speed and chamber pressure and their phase discrepancy allows for the determination of the unsteady response of the solid propellant combustion to pressure oscillations.

Ongoing research is focused on the improvement of that bench [95]. In the future, it could be used to obtain the pressure response of the propellant, which could provide additional constraints for the optimisation of the model parameters.

### Temperature sensitivity measurement

Currently at ONERA, no test bench is available to measure the dependence of the steady-state surface deflagration rate with respect to the initial propellant temperature  $T_0$ . The determination of this dependence would lead to a more complete characterisation of the propellant. Note that the ZN approach discussed in Section 2.4 requires the knowledge of this sensitivity. It could be obtained with a modified closed bomb, where the initial temperature is carefully controlled. Performing multiple tests at various values of this temperature enables to determine the sensitivity coefficient  $\partial_{T_0} \log(m)$ . The reproduction of that sensitivity with the 1D model could be used as an additional constraint in the optimisation procedure described in the following sections.

## 9.2 From experimental data to model parameters

Using the previously presented test benches, we are able to characterize a given composition by its thermal properties, its steady burn rate and its ignition delay. We aim at constructing a numerical model that reproduces those results with good accuracy. To that end, the various model parameters must be linked to the experimental data. We have already seen that certain parameters, e.g. solid phase parameters such as  $\lambda_c$ ,  $c_c$ ,  $\rho_c$ , and the surface emissivity can be easily obtained from simple inert tests. In the following, we explore how some other parameters can be deduced from thermochemical considerations. Still, a few parameters cannot be directly identified. We thus propose to use them as optimisation variables in a numerical fitting process presented in the next section.

### 9.2.1 Gas phase parameters obtained from equilibrium considerations

For the gas phase, we keep the global reaction mechanism from our early model. We need to specify the properties for both species  $G_1$  and  $G_2$ . The species  $G_2$  represents the combustion products. The equilibrium code COPPELIA [293] (Calculs et OPTimisations des Performances Énergétiques des systèmes Liés à l'Autopropulsion) developed at ONERA allows to compute, based on the precise solid phase chemical composition, the equilibrium composition of the mixture of combustion products, its equivalent molar mass  $\mathcal{M}$ , dynamic viscosity  $\mu$ , thermal conductivity  $\lambda$ , Prandtl number  $Pr$ , specific heat  $c_p$  and flame temperature  $T_f$ . This code assumes that all gaseous species behave like perfect gases, and that the combustion is completed at constant pressure in an adiabatic chamber, which is reasonable for large-scale SRMs. It assumed that the specific volume of potential solid products is negligible.

We further assume that the burnt mixture has a unity Schmidt number, and we retain the constant Lewis, Schmidt and Prandtl numbers assumption from the generic model. Finally, we assume that the global reactant species  $G_1$  (pyrolysis products) is identical to the combustion

products  $G_2$ , except for its formation enthalpy  $h_{G_1}^0$ . This modelling choice is approximate, however it leads to a great simplification of the simulation process, while still producing good results.

### 9.2.2 Further parameter relations and bounds

A few model parameters or model outputs which can be linked to experimentally obtained data. From a set of closed bomb firings, the steady-state burning rate law  $r = aP^n$  can be evaluated, yielding a value for the pressure exponent  $n$ . For single reaction chemical mechanisms, it is shown theoretically in [156] that this exponent is close to  $n_r/2$ , with  $n_r$  the reactant concentration exponent in the associated Arrhenius reaction rate law. Thus a very sensible initial guess for this concentration exponent is obtained, and we will be able to let our optimisation procedure refine its value if necessary.

The steady-state surface temperature is not measured at ONERA, however we assume it lies within the range [750, 1000] K, typical of aluminised propellants [294]. No direct formula can link this temperature to the other model parameters, hence this requirement will translate into two inequality constraints for our optimisation procedure.

A useful relation is obtained by considering a global energy balance:

$$T_f = T_{std} + \frac{h_c^0 + c_s(T_i + T_{std}) - \sum_{k=1}^{n_e} Y_{k,f} h_k^0}{\sum_{k=1}^{n_e} Y_{k,f} c_{p,k}} \quad (9.3)$$

The formation enthalpies only need to be defined relative to that of the solid phase  $h_c^0$ . We thus set the latter to 0 J/kg at  $T = T_{std} = 0$  K. Also, in the case of our simplified modelling (2 species with identical specific heats, one irreversible reaction), further simplification of the previous equation are possible. Overall we obtain:

$$T_f = \frac{c_s T_i - h_{G_2}^0}{c_p} \quad (9.4)$$

Thus, knowing  $T_f$ ,  $T_i$ ,  $c_s$  and  $c_p$ , we can deduce the value of the formation enthalpy  $h_{G_2}^0$  for the combustion product  $G_2$ . The last remaining formation enthalpy to be prescribed is that of the pyrolysis product  $G_1$  which is  $h_{G_1}^0$ . The phase transition at the surface is endothermic. However the solid phase degradation and decomposition processes can be endothermic or exothermic. This information and the value of the associated heat of reaction is usually poorly known and has been debated for decades in the case of classic AP-HTPB propellants [33]. Following Equation (3.1), the formation enthalpy  $h_{G_1}^0$  is related to this unknown heat of reaction. Thus, we do not impose a precise value, but we instead let our optimisation procedure iterate on this value freely, within a physically acceptable range.

Finally, since we impose a constant Schmidt number, the species diffusion coefficient  $D$  is easily obtained as:

$$D = \frac{\mu}{\rho \text{Sch}} = \frac{\mu RT}{P \text{Sch}} \sum_{k=1}^{n_e} \frac{Y_k}{\mathcal{M}_k} \quad (9.5)$$

### 9.2.3 Free parameters

The remaining model parameters thus form the vector of variables to be optimised. They are listed below:

- parameters for the pyrolysis law  $A_s T^{\beta_s} \exp(-\frac{T_{as}}{T_s})$ :
  - pre-exponential factor  $A_s$
  - temperature exponent  $\beta_s$
  - activation temperature  $T_{as}$
- parameters for the chemical reaction rate  $A_g T_g^\beta [G_1]^{n_r} \exp(-T_{ag}/T)$  of the single irreversible gas phase reaction:
  - reactant concentration exponent  $n_r$
  - preexponential coefficient  $A_g$
  - temperature exponent  $\beta$
  - activation temperature  $T_{ag}$

Note that  $h_{G_1}^0$  was also added to the list of optimised variables, however it has a negligible effect on ignition. It was therefore set to 0 at  $T = 0$  K for simplicity. Additional considerations should be used to settle its value. In total, we are left with 7 parameters to optimise, which we gather in our vector  $X \in \mathbb{R}^7$  of optimisation variables.

## 9.3 Optimisation framework

In the previous part, we have seen what data is available at ONERA to characterise a propellant composition. Inert and steady-state measurements provide many relations between the parameters of our simplified model, and some of these parameters can be directly deduced from the measurements. However there remains a few parameters which cannot easily be fitted manually. Our approach is to provide sensible initial guesses for the free variables and use an optimisation procedure. The latter improves these values via an iterative process that uses the VULCID code to test the parameter values and assess how well they reproduce the experimental results. Let us now explain how the optimisation is performed.

### 9.3.1 Definition of the optimisation problem

Apart from the experimental measurements which are explicitly related to some of our model parameters, the experimental results mostly consist of three information which cannot directly be used to determine our model parameters:

- the coefficients  $a$  and  $n$  of the steady-state burning law which describes the pressure dependence of the burning rate with respect to the gas phase pressure.
- the surface temperature evolution during laser ignition
- the ignition law fitted on these ignition measurements

As already discussed, the early temperature evolution during laser ignition is used to assess the effective laser heat flux based on the known thermal and absorbance properties of the solid phase. Trying to reproduce them directly with the 1D model would be impractical because of the non-uniformity of the surface temperature on the propellant sample. Instead, it is more practical to simply focus on the ignition times, which are determined in an average manner on the whole surface and are therefore less error-prone.

The lack of known relations to enforce the previously listed characteristics, and the nonlinear dependence of these on the optimisation variables lead us to consider the framework of constrained nonlinear optimisation.

From a mathematical point of view, we handle the steady-state burning rate law as an equality constraint. This is sensible, since we want our model to accurately reproduce steady-state combustion, so that the quasi-steady behaviour of an SRM simulated with that model is coherent. The corresponding equality constraints are defined as:

$$h(X) = \left( \frac{a(X) - a_{target}}{a_{target}}, \frac{n(X) - n_{target}}{n_{target}} \right)^t = 0 \quad (9.6)$$

where the functions  $a$  and  $n$  give the corresponding values of the Vieille law coefficient for the configuration obtained with the values from the optimised variables vector  $X$ . The subscript *target* denotes the experimentally obtained coefficients.

To ensure the physical coherence of the model, we also add an inequality constraint such that the steady-state surface temperature  $T_{ss}$  obtained at a specific pressure  $P_s$  stays within the reasonable bounds  $[T_{ss,min}, T_{ss,max}]$ . This is formulated as the following vector inequality constraint:

$$g_{T_s}(X) = (T_{ss}(X, P_s) - T_{ss,max}, T_{ss,min} - T_{ss}(X, P_s))^t \leq (0, 0)^t \quad (9.7)$$

Another set of inequalities is obtained by enforcing bounds on all variables. For instance, activation temperatures are not allowed to be negative. This also keeps the optimisation algorithm from straying too far away from physically realistic configurations, for which VULC1D could experience convergence issues. These bounds are formulated as:

$$g_{bounds}(X) = (X - X_{max}, X_{min} - X)^t \leq 0 \quad (9.8)$$

Overall the inequality constraints are gathered as a single vector function:

$$g(X) = (g_{T_s}(X), g_{bounds}(X))^t \leq (0, 0, 0, 0)^t \quad (9.9)$$

Finally, an objective function must be provided. Our goal is to reproduce experimental ignition times, therefore the following scalar objective function is defined. It is a positive function, which is only equal to 0 when the ignition law is perfectly reproduced:

$$f_{obj}(X) = (p_1(X) - p_{1,target})^2 + (p_0(X) - p_{0,target})^2 \quad (9.10)$$

with  $p_1$  and  $p_0$  the coefficient of the fitted ignition law (9.2).

Overall, the constrained optimisation problem can be written as follows:

$$\min_{X \in \mathbb{R}^N} f_{obj}(X) \quad (9.11a)$$

$$\text{subject to } g(X) \leq 0 \quad (9.11b)$$

$$h(X) = 0 \quad (9.11c)$$

To improve the conditioning of the problem and the convergence, as well as easing the numerical approximation of the Jacobians of  $f$ ,  $g$  and  $h$  with respect to  $X$ , each physical variable to be optimised is transformed before being inserted in the vector  $X$ . For instance, the pyrolysis law Arrhenius pre-exponential factor can vary between  $10^6$  and  $10^{12}$ . Thus it is replaced by its logarithm to base 10, which then varies in the more compact range [6, 12]. After applying the logarithm transformation if required, each optimisation variable is then rescaled between 0 and 1, such that 0 corresponds to the lower authorised bound, and 1 to the upper one.

### 9.3.2 Optimisation algorithm

The optimisation algorithm SLSQP (Sequential Least Squares Programming) from the Python library `Scipy` [239] is used to perform the constrained optimisation. This algorithm is especially efficient for solving constrained optimisation problems with nonlinear constraints. A very detailed review of the algorithm and its improvements is presented in [295]. It can be seen as a generalisation of the Newton and quasi-Newton methods for problems with constraints.

SLSQP is not a feasible-point method, i.e. there is no insurance that the iterates satisfy the constraints, unless convergence has been reached. This property is a strong advantage in the case of nonlinear constraints, as is the case of our burning rate law constraint. Indeed, ensuring the feasibility of every iterate would be much more complex, and would require a feasible initial solution, which is already impossible to define. Still, this algorithm has also been tweaked to ensure that simpler constraints, e.g. bounds on the optimised variables, are always satisfied. This avoids convergence or arithmetic problems in `VULCID` by inhibiting the use of unphysical values during the whole optimisation process.

The basic principle of SLSQP is described in the following. The objective function  $f$  is replaced by the Lagrangian:

$$\mathcal{L}(X, \lambda_{eq}, \lambda_{ineq}) = f(X) + \lambda_{eq}^t h(X) + \lambda_{ineq}^t g(X) \quad (9.12)$$

with  $\lambda_{eq}$  and  $\lambda_{ineq}$  the vectors of Lagrange multipliers for the equality and inequality constraints.

At the  $k$ -th iterate  $X^k$ , the complete nonlinear problem (9.11a) to (9.11c) is approximated as a quadratic subproblem, i.e. a second-order approximation around  $X^k$ :

$$\mathcal{L}(X^k + d, \lambda_{eq}^k, \lambda_{ineq}^k) \approx \mathcal{L}(X^k, \lambda_{eq}^k, \lambda_{ineq}^k) + \nabla \mathcal{L}(X^k, \lambda_{eq}^k, \lambda_{ineq}^k) d + \frac{1}{2} d^t H_{\mathcal{L}}(X^k, \lambda_{eq}^k, \lambda_{ineq}^k) d \quad (9.13)$$

with  $H_{\mathcal{L}}$  denoting the Hessian of  $\mathcal{L}$ , and  $d$  is the variation of the input vector around  $X^k$ . In practice, the Hessian is not computed directly, as its numerical approximation may be cumbersome and costly to obtain. Instead, it is gradually approximated, following the BFGS approach [296]. Starting from an initial guess (typically the identity matrix) for  $H_{\mathcal{L}}$ , the optimisation initially behaves like a simple gradient-descent method, i.e. taking steps in the direction where  $f$  diminishes most at first-order. The Jacobian of  $\mathcal{L}$  can be evaluated at each successive iterate and, assuming its variations are due to the second-order term involving  $H_{\mathcal{L}}$ , an approximation of the latter can be refined gradually. For each SLSQP step, the quadratic subproblem is constrained by linear approximations of the inequality and equality constraints around the previous step, and the Lagrange multipliers  $\lambda_{eq}$ ,  $\lambda_{ineq}$  are fixed at their values from the previous step. After the subproblem is solved, the new iterate  $X^k$  is obtained, and the process is repeated until convergence. This way, the optimisation improves the objective function and lowers the constraint violations progressively. Convergence is reached once the norms  $\|X^{k+1} - X^k\|$  or  $\|f(X^{k+1}) - f(X^k)\|$  are below user-specified tolerances.

Note that there is no guarantee that the solution is the global optimum of the problem. It could be envisioned to perform multiple optimisation with different starting points, or on separate intervals to obtain a better picture of the overall shape of the objective function, for example with the latin hypercube sampling approach [297].

This optimisation framework enables the optimisation of the model parameters, ensuring nonlinear relations are satisfied by the associated configuration (e.g. burning rate law). Such a framework is a very useful addition to the tools available at ONERA, since the model developed in this work presents a larger number of parameters than previous ignition models [153]. Only unconstrained optimisation were previously available [283]. To our knowledge, such an optimisation approach for a solid propellant model has not been presented in the literature.

## 9.4 Results

The SLSQP algorithm is executed on the previously described optimisation problem. Convergence is achieved when the objective function  $f$  varies by less than 0.1% between two consecutive iterates. This parametrisation yields very good results within a few iterations, typically 5 to 20. Sensitivity studies of the result to the initial solution and to various numerical aspects (e.g. Jacobian accuracy) have not been conducted, but should be considered in the future.

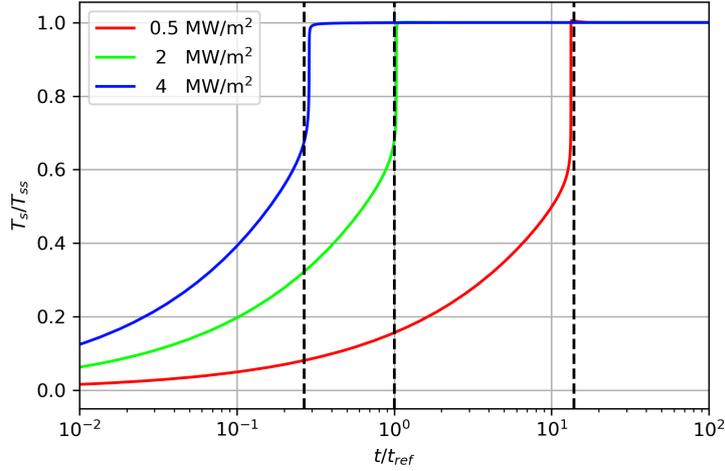
We have observed that the slope  $p_1$  of the ignition law was the most difficult experimental result to reproduce. This slope is mainly related to the pyrolysis law activation energy  $T_{ap}$ , however the optimisation algorithm had trouble with the convergence of this value, which may be linked to opposite effects of  $T_{ap}$  on  $p_0$  and  $p_1$ . Therefore multiple optimisations have been performed with various fixed activation temperatures, and the best result is selected manually based on the quality of the numerical ignition law obtained. Each fit satisfies with great precision the steady-state burn rate law, and the steady-state surface temperature lies within the prescribed range as expected. The quality of the final result is shown visually in Figure 9.8, where the evolution of  $T_s$  is shown for 3 laser flux levels, typical of the wall heat flux encountered in real motors. The vertical lines indicate the target ignition time as prescribed by the experimentally obtained ignition law. We clearly see that the numerically obtained ignition times are very close to those. We thus have great confidence in the ability of the model to reproduce the ignition dynamics in a large-scale 3D simulation of the ignition of a complete motor.

### 9.4.1 Optimised parameters

The convergence point can then be used to generate the optimised model. For the solid phase, its properties are:  $\rho_c = 1800 \text{ kg.m}^3$ ,  $\Delta h_f^o(c) = 0 \text{ J/kg}$  at  $T = 0 \text{ K}$ ,  $c_c = 1150 \text{ J/kg/K}$ ,  $\lambda_c = 0.55 \text{ W/m/K}$ . The pyrolysis law is  $m_s = A_p T_s^\beta \exp(-T_{ap}/T_s)$ , with  $A_p = 1 \times 10^9 \text{ kg/s/m}^2$ ,  $\beta = 0.3$  and  $T_{ap} = 1.5 \times 10^4 \text{ K}$ .

The reactant  $G_1$  and product  $G_2$  have the same properties except standard enthalpies. Their molar mass is  $\mathcal{M} = 27 \text{ g/mol}$ , and their heat capacity is  $c_p = 2800 \text{ J/kg/K}$ . The standard enthalpies at  $T = 0 \text{ K}$  are  $\Delta h_f^o(G_1) = 0 \text{ J/kg}$  et  $\Delta h_f^o(G_2) = -8.9 \times 10^6 \text{ J/kg}$ . The diffusion coefficients are equal for both species and taken as a linear function of  $T$  such that the Schmidt and Prandtl numbers remain constant (1 and 0.5 respectively). The thermal conductivity is  $\lambda = 0.45 \text{ W/m/K}$ .

The unique global reaction  $G_1 \rightarrow G_2$  is irreversible. The reaction rate is computed as:  $\omega = A[G_1]^n T \exp(-T_a/T)$ , with  $A = 2750 \text{ s}^{-1}$ ,  $T_a = 1500 \text{ K}$ ,  $n = 0.6$  and  $[G_1]$  the concentration of  $G_1$ .



**Figure 9.8** Surface temperature evolution for 3 different heat flux levels and expected ignition times (vertical dashed lines) based on the experimental ignition law

### 9.4.2 Sensivity of the propellant characteristics

At the convergence point, sensitivities of the various model characteristics can be evaluated. They are gathered in Table 9.1. Each line corresponds to the sensitivity of one characteristic with respect to the model parameters that have been optimised. The values are taken as relative variations of the characteristic value with respect to a relative variation of variable. For instance, the first element of the first line is:

$$\frac{\partial(a/a_{cv})}{\partial(A_p/A_{p_{cv}})}$$

with the subscript *cv* denoting values obtained at convergence. In the table, the highest absolute sensitivities for each characteristic are written in bold. We clearly see that one variable usually dominate all the others in terms of sensitivities, except for the case of  $T_{ss}$  where all sensitivities are of the same order of magnitude.

	$A_p$	$T_{ap}$	$\beta_p$	$n_r$	$A$	$T_a$
$a$	2.3e-08	-6.0e-07	2.4e-08	<b>-4.7e-06</b>	8.8e-07	-7.1e-07
$n$	4.8e-06	-4.1e-05	5.7e-06	<b>2.8e-03</b>	-8.4e-07	3.6e-05
$T_{ss}$	-3.6e-01	<b>7.3e+00</b>	-3.9e-01	4.4e-01	1.8e-01	-1.3e-01
$p_1$	-5.2e-06	<b>8.5e-04</b>	-9.0e-07	4.3e-08	1.7e-06	-2.6e-06
$p_0$	5.7e-04	<b>-1.5e-02</b>	5.9e-04	-4.8e-07	1.0e-06	-1.6e-06

**Table 9.1** Relative sensitivities of various characteristics with respect to the optimised model parameters at convergence

## Partial conclusion

In this chapter, we have presented the experimental benches in activity at ONERA for the characterisation of solid propellant combustion. The experimental data allows for the evaluation of

the thermal properties of the solid propellant, the steady-state combustion and its sensitivity to pressure, as well as the ignition dynamics of a given composition.

Starting from the global model from Section 3.3.3, some model parameters can be directly linked to the measured characteristics. However, properties such as the burning rate law and the law of ignition delay cannot be explicitly formulated in terms of model parameters. Thus, a numerical approach is developed, where some parameters are optimised so as to approach these properties. It has been chosen to impose the burning rate law as an equality constraint, i.e. it must be very precisely matched, so that the quasi-steady combustion behaviour of the model is coherent. The ignition law is then the target property which must be approximated as closely as possible. Very good results have been obtained, with an excellent agreement between the simulation and experimental results for laser-induced ignition.

Improvements can be made to the present methodology by exploring in a systematic manner the solution space. For that purpose, multiple optimisations should be performed, starting from various initial solutions that scan the acceptable parameter space in an adequate manner.

Overall, this chapter has shown the adequacy of a constrained optimisation approach to fit model parameters to experimental data. When available, additional results such as surface temperature, sensitivity to initial temperature, frequency response should be used to provide additional constraints to the problem and increase the realism of the optimised configuration.

The set of parameters generated here will be used in Chapter 11 for the simulation of ignition in a combustion chamber.



# Chapter 10

## Code developments

### Summary

We present the multiple code developments that have been conducted during the course of the PhD. The global code structures are described, and some technical choices are discussed.

In the previous chapters, we have presented several numerical tools: the semi-analytical solver, the one-dimensional solver `VULC1D` with the `ESDIRK` time integration as well as its Python prototype, the parameter fitting tool for the 1D ignition model, the optimisation tool for the creation of unstable configuration, the integration of `VULC1D` as a dynamic boundary condition in `CEDRE` (1D flame and detailed modelling approaches), the adaptive coupling demonstrator. Also, a wide variety of post-processing scripts have been used to analyse the simulations results and to perform parametric studies automatically.

All these tools have required a number of code developments, which form a large part of the work accomplished during the three years of this PhD thesis. We believe it is essential that the description of such developments be included in the manuscript, as well as the related technical and algorithmic choices. Therefore, the present chapter provides an overview of the various codes and their implementation.

Computationally heavy numerical tools have been implemented in Fortran, while early prototypes and analysis tools have been developed in Python.

## 10.1 Standalone solvers

### 10.1.1 Semi-analytical tool

The semi-analytical resolution method presented in Chapter 3 enables steady-state solution profiles to be computed for the model from Chapter 2, in the case where only 2 species are considered in the gas phase. The low-cost shooting method from Section 3.3 lends itself quite well to a Python implementation, taking advantage of the various root-finding methods and temporal integration schemes available in `Scipy`.

**Solution method** We have seen in Chapter 3 that the steady-state solution is found by iterating on the values of the regression speed  $r$  until the point where the interface thermal balance mismatch

is zero, i.e.  $\xi(r) = 0$ . We have 2 bounds  $r_{min}$  and  $r_{max}$  on  $r$ , and we have shown that  $\xi$  is monotonous function of  $r$ . Thus, any root-finding method can be used to iterate on  $r$ .

The remaining difficulty comes from the determination of the value of  $\xi$  for a given regression speed. This requires solving the solid and gas phase temperature Equations (3.26) and (3.27). The solid phase thermal profile can be directly integrated analytically. The gas phase temperature profile, however, must be integrated numerically. The various methods of the `solve_ivp` package from `Scipy` have been tested, however only LSODA [276, 277], was able to achieve tight relative error tolerances smaller than  $10^{-6}$ . A Python interface, developed by CMAP at École Polytechnique, to the Fortran code `Radau5` from Hairer and Wanner [206] has also been tested and performs at least as well and is more robust overall, hence this method is retained as the default choice. It is also the only method that could satisfy the error tolerance with  $rtol = 10^{-12}$  to ensure that no error is coming from the gas phase solution.

Once the thermal profiles from both phases have been obtained, the surface imbalance  $\xi$  can be computed.

**Code structure and performance** To provide a convenient interface, a Python object has been written, which holds the required model parameters and all the necessary functions: model function for the gas phase, integration of the gas phase temperature profile, computation of  $\xi$ , root-finding for  $\xi$ , solution export, linear stability analysis by the ZN method [30]. The parameters are passed as a dictionary to the object.

The computation of  $\xi$  is encompassed in a Python function, which is passed to a scalar root-finding method. Various methods are available in the library `scipy.optimize` such as the bisection method, Newton's method or Brent's method [207]. The latter has been retained as default choice due to its superior performance in practical cases. It dynamically switches between the bisection, secant, and quadratic inverse approaches, enabling a faster convergence. With that approach, 10 iterations are usually required to obtain the steady-state regression speed  $r$  within a  $10^{-14}$  relative error tolerance, each iteration requiring one numerical integration of the gas phase temperature profile.

The typical overall solution time is 0.3 s to 2 s depending on the nonlinearity of the gas phase source term. The steady-state CFD Fortran solution program, which is a version of `VULC1D` specialised for steady-state used in Section 3.3, required a similar time to obtain a converged steady-state solution. However, it must be noted that this code starts from a sensible user-defined initial guess. Therefore, even with the performance disadvantage of the Python implementation, the semi-analytic tool remains competitive, while being much more robust and easy to use in practice, since no initial solution and mesh are required. Also, the achieved precision on the regression speed and temperature profile is much higher, even though the relative error achieved with the Fortran code on the regression speed  $r$  is already very good at approximately  $10^{-7}$ , see Figure 3.4.

### 10.1.2 One-dimensional unsteady propellant solvers

The unsteady model from Chapter 2 has been studied extensively in Part I, and a numerical strategy for the temporal integration of the corresponding semi-discrete equations obtained with a finite volume approach has been presented in Chapter 5. Originally, before contemplating high-order integrators, a solver with first-order accuracy in time has been developed from scratch in Python. It has served as an experimentation basis for the exploration of more advanced temporal integrators. Finally, the `VULC1D` Fortran code has been developed from an existing steady-state code from

ONERA, which has for example been used for the study of aluminium droplet combustion [112], incorporating all the developments successfully tested in the Python prototype.

## Python prototype

The initial focus has been on the development of a standalone gas phase solver. For instance, the implementation of the semi-discrete equations obtained with a finite volume-scheme has been constructed from scratch. A function returns the residuals corresponding the set of DAEs obtained after discretisation in time via backward Euler, returning a vector of  $n_g(2 + n_s)$  components, i.e. for the 2 variables ( $T$  and  $m$ ) and  $n_s$  species in each gas phase mesh cell. Each of these components corresponds to the residual, i.e. difference between the RHS and LHS, of one of the conservation equation obtained after discretisation in space and time, for the given mesh cell.

The solution components are stored within the vector  $X$ . At each time step, going from time  $t_n$  to  $t_{n+1}$ , a nonlinear system of the form  $F(X_{n+1}) = MX_n + F(X_{n+1})$  needs to be solved. This is accomplished by a quasi-Newton method, also implemented in Python. Each Newton step is computed by a forward-backward substitution, using the LU-framework available in `Scipy`, which is simply wrapping standard LAPACK functions.

The solid phase solver has then been introduced in a similar manner, with its own residual function, which returns a vector of  $n_c$  components, one for each solid phase mesh cells. The representation of the interface coupling conditions has also been implemented in a separate function, which returns a vector of  $2 + n_s$  components, one per physical field.

The coupling of the solid and gas phase is then accomplished by solving the nonlinear system whose residuals are the concatenation of those produced by the 3 previous functions. Thus, a fully coupled resolution scheme is constructed, which is first-order accurate in time.

After this initial development, the higher-order ESDIRK methods have been implemented. These require the storage of the time derivatives at each Runge-Kutta stage, which is done with preallocated matrices. The nonlinear system to be solved is essentially the same, the only real difference being the addition of constant terms which represent the action of the previous stages. Their implementation has consequently been relatively straightforward. In particular, any code using implicit Euler or BDF as original solution method can, in practice, be easily adapted to use ESDIRK methods instead, as discussed in Section 5.1.6.

**Code structure and performance** The description of each phase and their corresponding time derivatives are implemented as functions stored within phase-specific packages. A coupling package gathers them and construct the coupled residuals that corresponds to an ESDIRK Runge-Kutta stage. The various phases communicate via their respective boundary conditions, which are updated by the coupling package at each call of the coupled residual function.

The model parameters are stored within a Python dictionary with a clear structure, explicitly separating each phase.

Simulations of ignition transients have been conducted successfully, yielding identical results as those obtained with the Fortran code `VULC1D` in Section 6.2. Differences in performance with the Fortran code are discussed in the next Section.

## Vulc1D

After the successful developments of high-order temporal methods in the Python prototype, a similar Fortran90 code maintained by Dmitry Davidenko has been retrieved. It originally used a backward

Euler or Crank-Nicolson scheme for time advancement but was primarily focused on the generation of steady-state solution profiles. The same conservation equations are considered, however a more complex modelling was already implemented for the gas phase (temperature-dependent properties, mixture-averaging, interface with Chemkin-like configuration files for the gas-phase kinetics). With the aim of porting the Python developments in Fortran to obtain a high-performance sequential tool, a large part of the original Fortran code has been rewritten.

**Code refactoring and technical choices** Data structures have been introduced to provide a clear separation between the surface, the solid and gas phases. Residuals of the coupled system of DAEs were originally constructed in a monolithic manner. To decrease the code complexity, improve its readability and allow for easier future developments, the residuals of each phase have been split in separate functions, similarly to the Python prototype.

The ESDIRK temporal discretisation has then been introduced in a similar manner as for the Python prototype. Several matrices are allocated for the storage of the temporal derivatives obtained at each Runge-Kutta stage.

The original code used a damped quasi-Newton algorithm, whose pseudocode is given in Algorithm 2, with native handling of the block-tridiagonal residual Jacobian. The term “quasi” refers to the fact that the Jacobian of the nonlinear system of residuals is not updated at each Newton step, and can even be kept constant across multiple time steps if convergence remains acceptable.

This algorithm has been tailored to improve its behaviour. For instance, the handling of convergence issues has been improved, so as to avoid the appearance of floating point errors which can appear when the residuals Jacobian is poorly conditioned, which may occur when the time step is too large. Early detection of such events triggers an automatic refusal of the current step and the time step is consequently diminished. In the case of a coupled computation with CHARME, there have been instances of diverging Newton steps, e.g. the Newton algorithm would let the temperature change by several hundred degrees in one iteration, clearly indicating a poor conditioning of the Jacobian. This resulted in a non-convergence which would stop the coupled simulation. The issue was circumvented by monitoring the relative size of the Newton increment for each variable. If a change too large is detected (typically more than 50 %), the step is refused and smaller time steps are taken. This has greatly improved the robustness of the coupled computations. Interfacing VULC1D with CHARME is discussed later in this chapter.

Comparisons have been made with an undamped Newton, obtained by simply inhibiting the damping step in the previous algorithm, and no sensible performance or robustness difference has been detected, even in strongly nonlinear situations, such as the imposition of a large pressure step with the detailed physico-chemical model from Section 6.4. This is coherent with the fact that all standard implicit solvers for stiff ODEs or DAEs, such as BDF and Radau [206, 218], use an undamped quasi-Newton method, while still being efficient on stiff nonlinear problems. Only under unrealistic conditions, such as applying a strong laser heat flux (10 MW without ramp-up), has the damped Newton clearly been superior to the undamped alternative, which struggled to converge. Since no performance loss was caused by the inclusion of the damping strategy, all the computations presented in this manuscript have been performed with the damped Newton.

A convergence problem appeared when considering reaction mechanisms where some concentration exponents are smaller than 1, as typically required to achieve the correct pressure dependence when considering only one global reaction. Indeed, when the corresponding reactant mass fraction tends to 0, the partial derivative of the reaction rate with respect to that mass fraction diverges. Therefore, zones with negligible concentrations in reactants after the main reaction zone would

cause permanent convergence issues with the damped Newton algorithm and resulted in a failure to complete the simulation. This issue has been circumvented with the help of Dmitry Davidenko by letting the corresponding concentration exponents tend to 1 as the associated reactant concentration tends to 0. The transition occurs below a fixed threshold for the mass fraction, which was set sufficiently low ( $Y_{\text{f}} \approx 10^{-3}$ ). It has been verified that the overall solution is not impacted by this smoothing. Note that this transition mechanism is not used for realistic complex chemical mechanisms, where the concentration exponents cannot be smaller than 1.

The alternative instantaneous mass flow rate constraint given by Equation (5.48) has also been implemented. This has shown that using the Runge-Kutta quadrature for the gas phase continuity equation is always more efficient than this instantaneous constraint, sometimes by a large margin ( $> 30\%$ ).

To aid with the development, a number of Python scripts have been written, for instance for the analysis of the Jacobian structure, which helped to spot bugs and issues. Also, the Fortran code has been modified to read initial solution profiles from a CSV file, which can be generated conveniently with Python. Overall, a large number of parameters has been introduced to fine-tune the behaviour of the numerical method: convergence criterion, safeguards against diverging Newton steps, time step adaptation parameters...

**Performance** The final tool is very robust, and can handle the simple model from Section 3.3.3 efficiently, as well as much more complex models as in Section 6.4. The new time adaptation capability and the high-order temporal resolution have resulted in a highly increased overall performance for engineering applications, by ensuring a precise resolution of the dynamics without relying on a prohibitive CFL-criterion. We recall that, in Section 6.4, the practical computational times were shortened from roughly 600 s ( $\Delta t$  set such that  $\text{CFL} < 10$ ) to 11 s only (adaptive integration with  $\text{rtol} = 1e - 5$ ). Typical gains in other use cases may be lower, however there is definitely a large gain in terms of engineering time, because the time step is dynamically adapted to ensure a given solution accuracy, thus leaving no doubt to the user on whether the temporal resolution was adequate or not. In the latter example, setting  $\text{CFL} = 10$  would typically not be assumed to produce a decent result, yet the adaptive method shows that highly accurate results can be obtained with even larger CFL numbers.

It is interesting to compare the performance of the Fortran and Python implementations of the 1D solver. With ESDIRK-54A and a relative error  $\text{rtol} = 10^{-6}$ , the ignition transient from Section 6.2, with a slightly coarser mesh (200 cells in the gas phase, 51 in the solid phase), the Python code takes 230 seconds to compute the solution, with 300 time steps and 250 Jacobian evaluations. The fully optimised version of the Fortran code VULC1D solves the same ignition transient in 2 seconds, with roughly the same number of steps and Jacobian updates. This represents a speed-up of two orders of magnitude over the Python prototype. Even if the latter is hindered by an unoptimised handling of the Jacobian matrix, this difference in performance clearly shows the advantage of using a high-performance compiled language.

These successful developments have inspired the proposal of an internship at the laboratory EM2C (Énergétique Moléculaire et Macroscopique, Combustion) from Centrale Supélec, instigated and supervised by Ronan Vicquelin. The aim was the implementation of ESDIRK solvers for the counterflow flame solver based on the AGATHE and COMMCOMB codes from EM2C, as reported in [298]. Further developments are however necessary and are currently in progress.

---

**Algorithm 2** Damped Newton

---

```
1: procedure NEWTON( $X_0, J_0$ )
2:   Inputs:  $X_0, J_0$  initial solution and jacobian
3:   Outputs:  $i_{err}, X$  return code and final solution
4:    $k = 0, \alpha = 1, iter = 0$ 
5:    $R_{old} = F(X_0)$  initial residuals
6:
7:   while True do Newton loop
8:      $iter + = 1$ 
9:      $\Delta X_{old} = J^{-1}R_{old}$  update Newton increment
10:     $\alpha = 1$ 
11:    while True do damping loop
12:       $k = k + 1$ 
13:       $X_k = X_{k-1} + \alpha\Delta X_{old}$ 
14:       $R_{new} = F(X_k)$ 
15:       $\Delta X_{new} = J^{-1}R_{new}$  updated Newton increment
16:      if  $\|\Delta X_{new}\| < \|\Delta X_{old}\|$  then Newton increment diminishes
17:        go to 30
18:      end if
19:       $X_k = X_{k-1}$  revert to last accepted iterate
20:       $\alpha = \min(\alpha^2, \frac{1}{2})$  fast damping decrease
21:      if  $\alpha < 0.05$  then
22:        if  $J$  was updated at  $X_{k-1}$  then
23:           $i_{err} = 1$ ; return  $i_{err}, X_k$  no convergence can be achieved
24:        else
25:          update  $J$ 
26:          break restart full Newton loop
27:        end if
28:      end if
29:    end while
30:    if  $\|\Delta X_{new}\| < tol$  then
31:       $i_{err} = 0$ ; return  $i_{err}, X_k$  Newton has converged
32:    end if
33:    if  $iter > N_{max}$  then
34:       $i_{err} = 1$ ; return  $i_{err}, X_k$  too many iterations
35:    end if
36:     $R_{old} = R_{new}$ 
37:  end while
38: end procedure
```

---

## 10.2 Python frameworks

In the course of this PhD, there has been a need to perform large parametric studies, e.g. for the study of the convergence rates in space and time for the various solvers, or for the generation of the propellant frequency response to pressure oscillations. Also, the need to optimise our model parameters has required a proper wrapping of the Fortran code `VULC1D`, so that the optimisation algorithm can automatically call it as required to evaluate an objective function for instance. The developments performed in that scope are summarised in this section.

### 10.2.1 Automatic parametric studies

For the purpose of performing large parametric studies automatically, a Python framework was developed, which enables an automatic parametrisation and execution of the Fortran code. A template folder is first manually created once, which contains the necessary configuration files and a `VULC1D` executable. In the configuration files, all the parameter values that need to be changed during the parametric study are replaced by unique and explicit keys (for instance “`{solid_density}`”).

In the Python framework, a dictionary is constructed, which holds all the required parameter values for the Fortran code. For a given parametric study, an initial dictionary is formed using default values, tailored for the case that needs to be run. This dictionary can then be duplicated and modified as required for the parametric study.

For each configuration (one modified duplicate of the initial dictionary), a simulation needs to be run with the Fortran code. For that purpose, the template folder is copied to a temporary folder with a unique name. A mapping function takes the corresponding Python dictionary as input and replaces the previously mentioned keys in the template files by the corresponding dictionary values. Using the `os` or `subprocess` Python packages, the Fortran code can then be called from the Python script. The associated result files (time history of the solution, statistics) are then loaded in the Python program and stored in a data structure (dictionary or list) for further use. These operations (folder creation, parameter writing, Fortran call, loading of the result) are handled by a single Python class `SimRunner`. It is therefore straightforward to use it for other purposes, e.g. optimisation of model parameters optimisation as described in Section 10.2.2.

A useful advantage of this strategy is the ability to parallelise the various Fortran simulations. The handling of each configuration (dictionary generation and Fortran code call) can indeed be performed in parallel with the package `joblib`. Overall, this enables a large amount of simulations to be run at once. These can be run on a high-performance server, and outside office hours, greatly increasing the number of configurations that can be studied and making it very easy to iterate on the definition of the parametric study.

Once the simulation results have been loaded, the post-processing can be performed in Python (also in parallel if required), for instance the determination of the ignition time, or the amplitude of the mass flow rate oscillation during a sustained pressure oscillation. Finally, plots are generated and exported with the `Matplotlib` library.

For the purpose of being able to reuse or reanalyse previous studies, a JSON backup file is generated after each batch of simulations has been performed. There, a description of the parametric study and the locations of the associated simulation results are stored, enabling the study to be loaded back in the Python framework without launching simulations anew.

In the scope of the present manuscript, this Python framework has been used extensively, and has been the key element for obtaining all the convergence curves presented for `VULC1D` and for the

semi-analytical tool, for instance in Chapters 6 and 3. Developing such a framework is not a simple task, however it becomes very profitable as soon as a few sensitivity studies have been performed.

For very specific debugging or solution analysis, a graphical interface was also developed with `Matplotlib` to visualise the evolution of the solution profiles with time. This has been especially useful when investigating the issue previously mentioned with reaction exponents smaller than 1 in `VULC1D`.

## 10.2.2 Optimisation of model parameters

In Chapter 6, the initial configuration from Section 3.3.3 has been used as starting point to generate configurations with varying degrees of instability, using the semi-analytical tool to assess the steady-state sensitivities and linear stability. This has enabled the determination of a set of physical parameters that leads to a limit cycle, as studied in Section 6.3. In Chapter 9, a similar initial configuration is taken as starting point, and is iteratively improved so that experimentally measured ignition times and regression rates are correctly reproduced. Both these tasks have required the use of the constrained optimisation algorithm SLSQP, as already discussed in each specific chapters.

From a technical point of view, the same procedure as for the automatic parametric studies has been used. The initial configuration is described by a Python dictionary. The set of variables to be optimised is linked to the corresponding values in the dictionary. Thus, each time the optimiser asks for the value of the objective function, or for the values of the equality and inequality constraints, new simulations can be run with the updated configuration. For that purpose, the previous Python framework that wraps the execution of the Fortran code `VULC1D` has been reused.

The optimisation problem is implemented as a Python class. This class enables an easy setup of the problem, by providing procedures to evaluate the objective function, the equality and inequality constraints. Several interfaces need to be defined to communicate with the optimisation solver. Let us consider the case of the parameter fitting from Chapter 9.

**Objective function** The objective function requires the computation of the fitted ignition law (9.2) for a given value of the optimisation state vector  $X$  which contains the model parameters being optimised. The simple model used in `VULC1D` leads to values of the law coefficient  $p_0$  and  $p_1$  which do not depend on the laser flux level, but only marginally on the pressure. All experimental tests having been performed at the same operating pressure, the determination of these coefficients can be done by simply performing 2 ignition simulations with `VULC1D` and 2 different laser flux levels. The adaptive and high-order time stepping of `VULC1D` allows for very precise solutions to be obtained. For these 2 simulations, ignition times are computed as the time when the surface temperature diverges by more than 5% with respect to inert heating solution. The ignition law coefficients are then obtained in a straightforward manner by performing an affine fit of  $\log(t_{igni})$  as a function of  $\log(q_{laser})$ .

**Constraints** The constraint function requires the evaluation of the steady-state burning rate law. This is done in a similar manner, by performing two steady-state computations with `VULC1D`, for two pressure levels. To ensure the steady state is reached, the simulations start with an inert propellant heated up by a laser heat flux until ignition occurs. The simulations are then continued without laser until steady state is reached.

Typically, both simulations are performed at pressures  $P$  and  $P + \delta P$ , e.g. with  $\delta P/P = 10^{-3}$ . The coefficient  $n$  is then easily obtained by performing an affine regression of the logarithm of the

burning rate law (1.5):

$$\log(r) = n \log(P) + a \quad (10.1)$$

**Inequality constraints** The inequality constraint on the steady-state surface temperature  $T_{ss}$  only requires the evaluation of the numerically obtained steady-state surface temperature without laser at a prescribed pressure  $P_s$ . This can be done in a similar manner as in the constraint function.

**Jacobians** By default, the LSLQP algorithm computes the Jacobians of the previous functions by finite differences. However, we have observed that this would usually yield poor approximations of the Jacobians, even when tweaking the finite difference step size applied to all variables. This is due to the fact that some variables need to be perturbed more than others to obtain a numerically significant change in the VULC1D result. Therefore, the Jacobian estimation method is implemented in a customised manner as another function of the solver object. This function performs the finite-differencing with manually tailored perturbation step sizes for each optimisation variable (typically 0.5%).

**Interface to Vulc1D** To run an ignition or steady-state simulation of the configuration corresponding to a given value of the vector of optimised variables  $X$ , the VULC1D code must be properly setup and run. To this end, the previously presented Python wrapper for VULC1D is used. A function of the optimiser class maps the values of the optimisation state vector  $X$  to the dictionary that describes the model parameters for VULC1D, this dictionary is then passed to the Python wrapper which feeds the VULC1D executable with the corresponding configuration files.

After the simulation is run, the exported surface temperature and mass flow rate histories are loaded and returned as Numpy arrays to the calling function, which can then process these, e.g. determine an ignition time, or simply obtain the corresponding steady-state values.

**Parallelisation** To improve the execution speed of the overall optimisation process, the various 1D simulations required for the evaluations of the various functions and their Jacobians can be performed in parallel simply with the use of the Python library `joblib`.

## 10.3 Developments in Cedre

The coupling described in Chapter 7 has required interfacing VULC1D with the CFD toolchain CEDRE from ONERA and, in particular, its fluid flow solver CHARME.

### 10.3.1 1D flame approach

The VULC1D source code has been integrated in the CEDRE source code in the folder for the boundary condition referred to as CL13, which specialises in solid propellant ignition models.

We remind the reader that, in the context of a coupled CFD simulation, the 1D code is run sequentially on each processor for each boundary face that belongs to a propellant-type boundary. It is clear that each face model will have a different evolution, in particular the residual Jacobian of different faces will not be equal. To avoid recomputing the Jacobian at each coupling time step, the block-tridiagonal Jacobian of each model is stored in CEDRE. The diagonal blocks are stored within a 4-dimensional tensor of shape  $n_{vars} \times n_{vars} \times n_{cells,1D} \times n_{models}$ , with  $n_{vars}$  the number of physical

fields in the 1D model (2 plus the number of species  $n_s$ ),  $n_{cells,1D}$  the number of 1D mesh cells (identical for all 1D models), and  $n_{models}$  the number of propellant-type boundary faces. The lower and upper blocks of the Jacobians are stored in a similar manner. The solution vector  $X$  for each model is stored as a  $n_{vars} \times n_{cells,1D}$  matrix inside VULC1D. Thus, at the CEDRE level, the solution vectors of all boundary models are stored in a 3-dimensional tensor of shape  $n_{vars} \times n_{cells,1D} \times n_{models}$ . The solution vectors are initialised at the start of the simulation by reading a CSV file.

At the beginning of each coupled step, CHARME updates its boundary conditions. Therefore, the 1D models are run for one CFD time step  $\Delta t_{CFD}$  on each CPU-domain one by one. For each model, the previous solution vector and Jacobian are loaded back into the VULC1D data structure from the storage tensors.

As previously discussed for the 1D code, non-convergence of the damped Newton algorithm for VULC1D can occur, or even divergence of the Newton step due to poor problem-conditioning. In that case, VULC1D dynamically detects the issue and takes smaller time steps.

Once the boundary 1D models have been run, the propellant mass fluxes, flame temperature and composition and flow velocity are gathered and the corresponding boundary fluxes for CHARME are evaluated, as described in Chapter 7.

### 10.3.2 Detailed approach

In the case of the detailed approach, where the propellant flame is solved within CHARME instead of VULC1D, the process is essentially the same, however the 1D gas phase is not required any more. To maintain the 1D residual Jacobian structure, the gas mesh is made of 3 fictitious cells, which are associated with trivial residuals such that the corresponding physical values ( $T$ ,  $y_t$ ,  $m$ ) are the same as those found at the propellant surface. Then, to evaluate the surface fluxes that appears in the connection conditions with the propellant flame, the surface residual function of VULC1D calls a routine of CHARME which, based on the surface state and field gradients at the surface on the gas side, computes the corresponding Navier-Stokes fluxes. The gradients are estimated via first-order finite differences between the VULC1D surface state and the state at the center of the first CHARME cell of the CFD mesh above the surface. This procedure is called at each Newton step with the iteratively updated surface state. Thus, the influence of the CFD flow field from CHARME on VULC1D is implicit. Note that the reverse influence has so far been treated explicitly, which can degrade the stability of the coupling.

## 10.4 Demonstrators of co-simulation

In Chapter 8, we have demonstrated the application of co-simulation techniques to two test problems. The toy model from Section 8.2 and the waveform relaxation procedure from Algorithm 1 have been implemented in Python.

A function has been written to compute the time derivatives of the discrete temperature values at each mesh point of a single slab. The various boundary conditions considered (Neumann, Dirichlet, Balance) require the computation of a reconstructed surface temperature, which acts as a ghost point. The properties and polynomial predictors necessary for the determination of this ghost point value are held within a Python dictionary which is passed as input to the function.

The polynomial approximations are computed by Python objects (one per coupling variable), which are instances of the `predictor` class. This class provides a Newton interpolation/extrapolation

tion ability and handles in a user-friendly way the addition of new sampling points, discarding the oldest ones as required.

For the temporal integration of each slab, the adaptive solver LSODA [278] from the `Scipy` library is used with a tight error tolerance ( $rtol = 10^{-13}$ ), ensuring that the integration error is negligible compared to the error introduced by the polynomial approximation of the coupling variables.

Although not discussed in Chapter 8, the `solve_ivp` package also enables each slab solver to exchange their dense output (a high-order interpolant of their solution). These can be used instead of the polynomial approximation, leading to a true waveform relaxation approach which, if it converges, is free of any error, no matter which coupling time step is used. This possibility is however not representative of what is possible with a complex platform such as CEDRE.

The Fortran co-simulation framework used to couple VULC1D with the 0D chamber model in Section 8.5 is similar to the Python one. For the estimation of the error on the coupling variables, the `predictor` structure provides an easy-to-use interface, which takes the updated coupling variables and compares them with the predicted ones. Based on these values and degree of the polynomial prediction, the optimal time step can be computed, which would have produced an error equal to the specified tolerance. This information is then used to dynamically adapt the coupling time step. For the automatic selection of the prediction order, another interface is provided to compute the optimal time steps for each possible order, and returns the order that yields the largest time step.

## Partial conclusion

Numerous code developments have been conducted during the course of this thesis. The Python language and its ecosystem have been very profitable for the development of prototypes, the analysis of simulation results, the automation of parametric studies, and for interfacing Fortran codes with standard optimisation algorithms.

Although not applied directly to the present thesis, a variety of other tools have also been implemented, mostly in Python, to study different aspects. For instance, a Python library to handle explicit, diagonally-implicit and fully implicit Runge-Kutta methods has been developed and used in the scope of graduate courses. Compressible flow solvers in 1D and 2D have been implemented for familiarisation with finite-volume schemes. A 2D surface regression computation tool has also been created to compute the evolution of a burning propellant grain, and has for example been used to obtain Figure 1.6 in Chapter 1. Finally, the Radau integrator from the `solve_ivp` package of the Python library `Scipy` has been extended for the solution of DAEs up to index 3. All these additional developments can be accessed at the author's GitHub page [86].



# Chapter 11

## Numerical assessment of the coupled approaches on a newly designed test case for two-dimensional ignition

### Summary

A 2D configuration is designed, such that the detailed coupling from Chapter 7 can be used to solve all flow scales as well as the propellant flame and conjugate heat transfer with high accuracy. Comparison with the results obtained with the 1D flame coupled approach show that the 1D representation of the flame does not impact the ignition dynamics, thus legitimating the inclusion of the propellant flame in the CFD boundary condition. Sensitivities of the simulated ignition to the spatial refinement and the time step are investigated.

One of the thesis first aim is to produce a software framework capable of simulating ignition transients in SRMs. In our general introduction and in Chapter 1, we have shown that this required a multiplicity of developments, either mathematical or numerical. All the preceding chapters were oriented towards those. For instance, Part I presented the development of a one-dimensional unsteady solid propellant model and the associated numerical solver. In part II, we have discussed the coupling of this solver with the CFD code CHARME for the simulation of SRM chamber ignition. In particular, two approaches have been introduced:

- the one-dimensional flame approach from Section 7.1.2
- the detailed approach from Section 7.1.1.

The aim of the latter approach is to serve as a reference to verify the solution dynamics obtained by the 1D approach, which relies on a larger number of modelling approximations. Finally, Chapter 9 presented the optimisation process that we developed for the determination of the various model parameters to match experimental laser-induced ignition times.

We propose to compare both coupled approaches on a test case that is representative of SRM ignition induced by impingement of a hot jet onto the propellant surface. This chapter is of particular importance. It is the convergence point of all the work presented in the previous chapters. The simulations that will be presented rely on the numerical tools developed for the one-dimensional solid propellant model from Part I, and for its coupling with the CFD software CEDRE, as presented in

Part II. The parametrisation of the propellant model and gas phase reaction mechanism is obtained from the experimental data via the optimisation procedure presented in Chapter 9.

This chapter provides several contributions. The first one is the use of the detailed approach to simulate multidimensional ignition, which has so far never been reported elsewhere. The second one is the comparison of the 1D flame and of the detailed approaches. No direct comparison is available in the open literature between these two approaches, This second contribution is the core one, because it will lead to the verification of the one-dimensional and quasi-steadiness assumptions for the propellant flame used in the 1D flame approach. We will show that these assumptions have little effect on the obtained ignition dynamics, and that the 1D flame approach can be used to accurately compute large-scale ignition transients. This particular result is of paramount importance, as it gives a strong basis to all the ignition studies presented this far in the literature, which all rely (implicitly at least) on a 1D flame model, e.g. [53, 55, 56].

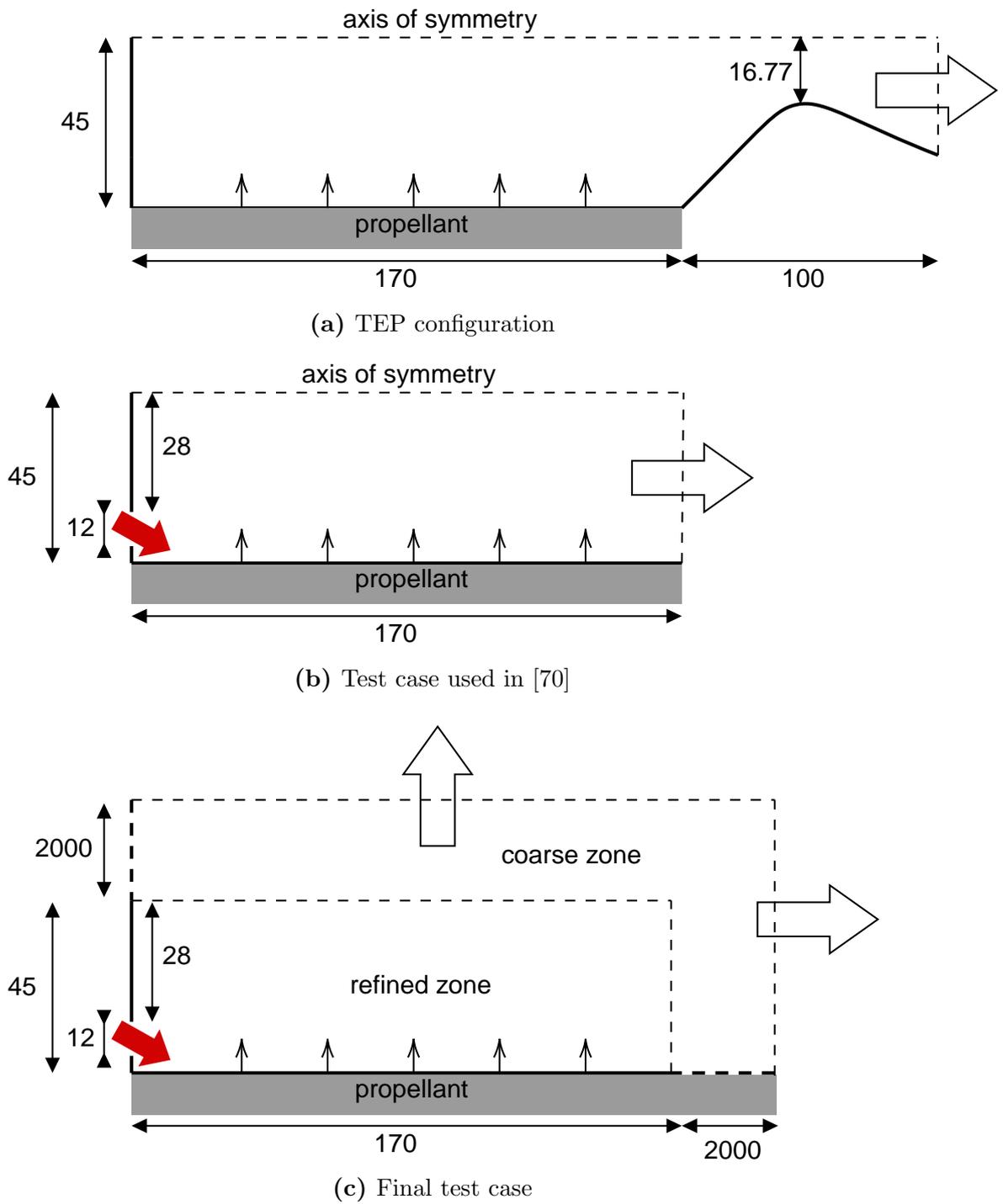
Finally, a third contribution arises from the comparison methodology and the configuration used. These have the potential to be far-reaching, by allowing more detailed studies of specific aspects of the modelling. For instance, the detailed comparison of both coupled approaches can enable the fine-tuning of various parameters or laws, such as the wall heat flux transition discussed in Section 7.3.2. Specific wall laws can also be generated with the detailed approach for the computation of the wall heat flux during the inert heating phase, with the aim of using a coarser mesh near the surface in practical computations.

An accurate comparison of both approaches can only be conducted if the observed differences in terms of dynamics can be attributed solely to the differences in surface propellant combustion models. In particular, all other phenomena, such as hydrodynamics and conjugate heat transfer, should be precisely resolved. To that end, the spatial discretisation of the computational domain and the time step used must be such that all temporal and spatial scales are properly resolved, so that the differences observed in the simulations with both approaches can be confidently attributed to the difference in the handling of the propellant flame and the surface connection conditions. This puts very stringent requirements on the CFD mesh refinement and time stepping. Therefore, to maintain an affordable computational cost, a simple configuration must be used.

Therefore, we start this chapter by presenting the genesis of the test case. Then, the numerical settings are discussed, and a reference simulation with the detailed approach is presented, with a very fine spatial mesh and a finely resolved temporal evolution. We then present simulations with the 1D flame approach on similar or coarser meshes, and compare them with the reference. Then, sensitivity to the time step used and to the near-surface CFD mesh refinement is numerically investigated. It is shown that, despite the sensitivity of the simulated dynamics to the surface model, results compare very favourably. The main conclusion is that accounting for the propellant flame only in the boundary model is acceptable and leads to accurate ignition dynamics.

## 11.1 Definition of the test case

We wish to design a configuration that is realistic in terms of dimensions, yet simple in terms of geometrical shape. The goal is to build a reference case that makes it possible to resolve all the various phenomena (combustion, hydrodynamics, conjugate heat transfer) with high numerical accuracy via extensive mesh refinement in the CFD domain, while maintaining an affordable computational cost, as is required for iterating on the definition of the configuration and for performing sensitivity studies easily. Then, comparing the 1D flame approach and the detailed approach for the solid propellant combustion, the discrepancies can be confidently attributed to the differences between



**Figure 11.1** Successive evolutions of the test case. Large arrows indicate subsonic outflow boundaries.

those two approaches, without a large impact from other sources of numerical errors, e.g. numerical dissipation. Due to the computational cost and mesh refinement associated with an accurate resolution of the propellant flame for the detailed coupled approach, but also the accurate capture of the conjugate heat transfer between the gas phase and the propellant, the configuration should be as simple as possible and, in particular, it would be preferable to have a two-dimensional setup so as to greatly reduce the size of the CFD mesh.

Let us make a brief overview of configurations available in the literature. Most 1D ignition models have been used directly on complex 2D or 3D SRMs [8, 52, 53, 56]. Also, fully documented experimental motors (geometry and propellants) are scarce in the literature. More refined physical models have been used by V. Yang and his colleagues for the study of transient combustion and acoustic instabilities in fictitious 2D rocket motors [299–302]. In these papers, the propellant flame is captured in the chamber fluid model. A very fine mesh is used, with cell heights of approximately 1  $\mu\text{m}$  near the surface, to accurately resolve the reactions zones and the thermal gradients. The solid phase is modelled with a one-dimensional heat equation, locally at each boundary CFD face. This approach is very similar to the detailed approach we introduced in Part II. Ignition was not considered in these studies, but only transient dynamics around the coupled steady-state solution.

If we are to simulate ignition with the detailed approach, we can expect that we will have to use similar grid refinements. To retain a manageable computational cost for a fully unsteady simulation, we cannot consider a fully three-dimensional simulation. Chinese researchers [56] have performed 3D simulations of the ignition of small SRM with a 1  $\mu\text{m}$  cell height at the surface, but a much coarser refinement in the longitudinal direction. The total number of cells was already on the order of 200 millions, which is far too large for the study proposed in this chapter. Therefore, we have to focus on a two-dimensional case.

We take inspiration from the fictitious axisymmetric motor TEP which has been used at ONERA for various studies, e.g. quantification of instabilities [303, 304], and as a simple verification case for new developments, such as gas phase radiation models [305]. The dimensions of the original TEP motor are close to those of tactical devices. The combustion chamber is 17 cm long, with an inner radius of 45 mm. The propellant grain is cylindrical and burns in a radial manner. Originally the motor possesses a nozzle with a throat radius of 16.77 mm, which is directly connected to the previous combustion chamber. The TEP motor is represented in Figure 11.1a.

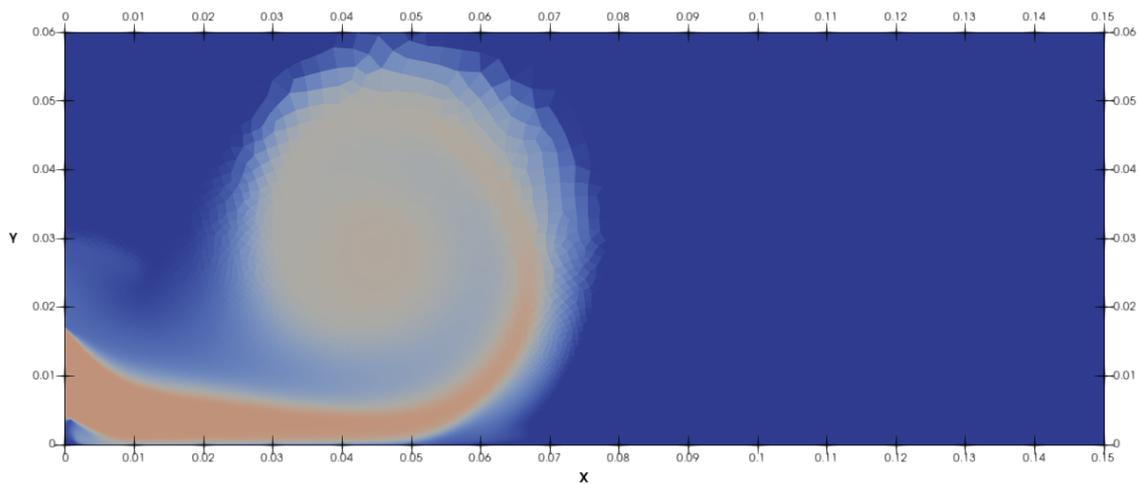
As we aim at studying ignition via impingement from a hot flow, we introduce an igniter in the aft-end of the motor, from which a hot jet flow enters the combustion chamber with an angle of 45 degrees with respect to the longitudinal axis, as represented by the red arrows. As in most SRM simulations reported in the literature, this igniter flow is assumed to be at chemical equilibrium, even though real igniter flows usually cannot be considered at equilibrium. We also assume that the combustion products of the igniter are the same as those of the main propellant load, but with a lower temperature. These choices allow for a reduction in the number of species and a simplification of the chemical pathways to be accounted for. The introduction of an igniter jet leads to a complex flow field which features flow instabilities and recirculation areas. This renders the test case very sensitive to the propellant surface dynamics. In particular, it can highlight differences in behaviours between various ignition models, making it a demanding test case, able to show discrepancies between the dynamics of different models for the solid propellant combustion.

To simplify the configuration, the nozzle is removed, and the nozzle entrance plane is replaced by a subsonic outflow condition. The resulting axisymmetric configuration has been used in one of our articles [70] and is sketched in Figure 11.1b. Although good first results were obtained in comparing both coupled approaches on this configuration, issues arose with pressure waves generated by the

start-up of the igniter jet. These reflect multiple times on both the axisymmetric and the right exit boundary conditions and come back to the surface of the propellant. We have observed that this phenomenon is the root cause of an exacerbated sensitivity of the overall simulation result with respect to the CFD mesh resolution. Solutions with slightly different mesh refinements drift slightly from each other after the first ignition, due to the intricate interactions between the flow field, acoustics, surface combustion, and numerical stability.

To avoid such issues, we have modified the configuration so as to discard these pressure-induced interactions. The 2D axisymmetric configuration is replaced by a planar 2D one, where the upper (formerly the symmetry condition) and right exit boundary conditions both correspond to subsonic outflow conditions and are pushed much further away from the propellant flame zone with a gradual mesh coarsening. Thanks to this modification, pressure wave reflections do not occur within the time range of interest, leading to a less disturbed unsteady solution. An alternative would have been to use non-reflecting boundary conditions, however their current implementation in CHARME does not function properly.

After a few iterations on this configuration, the final simplified test case is obtained and its details are summarised in this section. A sketch of the configuration is given in Figure 11.1c, and a snapshot of the temperature field during the initial igniter impingement is shown in Figure 11.2.



**Figure 11.2** Geometry and principle of the test case (temperature field at  $t = 2$  ms)

### 11.1.1 Geometry and boundary conditions

The fluid domain is 2D plane and rectangular. The left boundary is an adiabatic wall boundary, with an opening near the bottom, through which the igniter flow is injected at  $T = 2300$  K with a mass flow rate of  $110 \text{ kg.m}^2.\text{s}^{-1}$  and an angle of  $45$  degrees. The igniter jet and its orientation are clearly seen in Figure 11.2. The igniter injection hole starts at  $y = 5$  mm and is  $12$  mm long. The igniter flow is maintained during the whole simulation. The injected flow is turbulent and the associated scalars are  $k = 100$  J/kg, and  $\omega = 1.8 \times 10^4 \text{ s}^{-1}$ . The velocity in the core of the established igniter jet is approximately  $200$  m/s.

The right and upper boundaries (only visible in the sketch) are located at  $x = 2$  m and  $y = 2$  m respectively. They are subsonic outflow conditions, with a prescribed pressure of  $0.5$  MPa.

The lower boundary is a solid propellant boundary. Each of its faces is associated with one instance of the one-dimensional model, solved with the VULC1D code following the coupled approach described in Part II. The zone of particular interest is the portion of the surface propellant between  $x = 0$  m and  $x = 0.1$  m. When the surface is ignited, the combustion products are injected with zero turbulent kinetic energy ( $k = 0$ ) and  $\omega = 1.8 \times 10^3 \text{ s}^{-1}$ .

### 11.1.2 Initial conditions

The initial fluid state is uniform at 0.5 MPa,  $T = 293$  K, zero velocity and turbulent kinetic energy. The solid propellant thermal profile is also initialised at  $T = 293$  K. We use the same physico-chemical modelling as presented in Section 9.4. Initially, only combustion products  $G_2$  are present in the CFD and (if active) in the 1D gas domains. The igniter flow also injects the same combustion products. In the CFD domain, the reactant species  $G_1$  only appears near the surface when the detailed coupling is employed.

### 11.1.3 Spatial discretisation

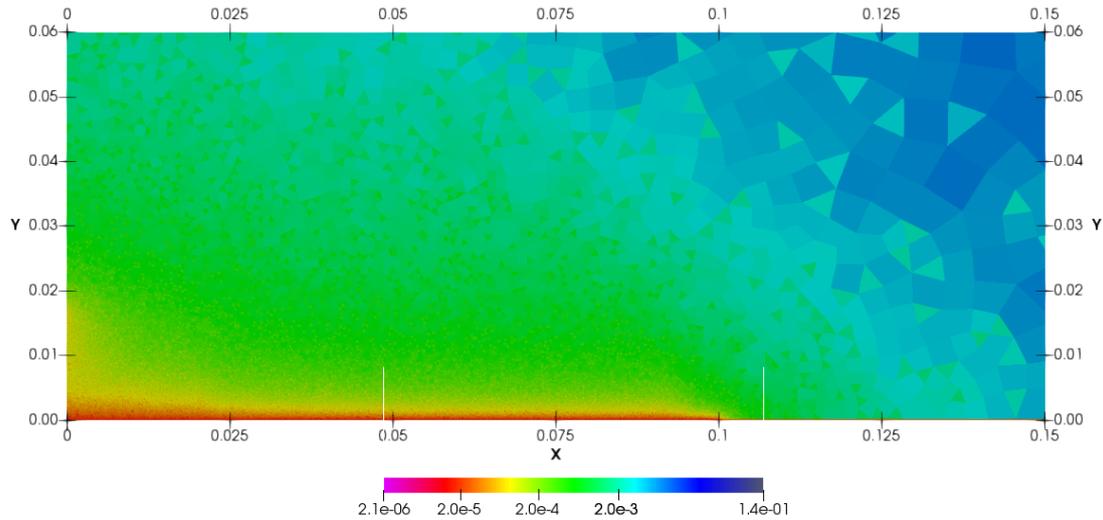
The domain is discretised in an unstructured manner with quadrangles and triangles using GMSH [306], with the exception of a small zone above the propellant, which is 200  $\mu\text{m}$  high in the  $y$ -direction and is meshed in a cartesian manner. This cartesian mesh allows for a fine control on the mesh size near the propellant flame. In particular, for the detailed approach, the flame must be captured within the CFD solver, hence the CFD mesh must be properly refined to capture the strong thermal gradients. A reference mesh is constructed where the first layer of cells above the propellant has a thickness  $\delta = 1 \mu\text{m}$  in the  $y$ -direction perpendicular to the propellant surface. In the structured part of the mesh, the cell thickness increases in a geometrical fashion as we move away from the propellant. The expansion ratio is close to 1, such that the mesh is smoothly coarsened, avoiding numerical issues arising from strong variations of the cell size. This meshing approach allows for a fine control of the near-surface spatial resolution, while preserving a fine unstructured mesh for the rest of the domain. Thanks to that, phenomena such as near-surface hydrodynamics, conjugate heat transfer and, in the case of the detailed approach, the gaseous propellant flame, can be resolved with a great spatial accuracy.

The field of equivalent diameter of the mesh cells is plotted in Figure 11.3 for this mesh. This diameter  $d$  is defined as:  $d = 2\sqrt{S/\pi}$ , i.e. it is the diameter of a disc that would have the same area  $S$  as the cell considered. The mesh is highly refined in the vicinity of the igniter flow and near the surface, so as to correctly resolve all the flow scales in this area.

Properly resolving the conjugate heat-transfer at the propellant surface requires a sufficient near-surface mesh refinement. This is generally verified with the criteria  $y_1^+ \leq 1$ , with  $y_1^+$  the dimensionless height of the first cell above the surface, defined as [245]:

$$y_1^+ = \frac{y_1 u_\tau}{\nu} \quad (11.1)$$

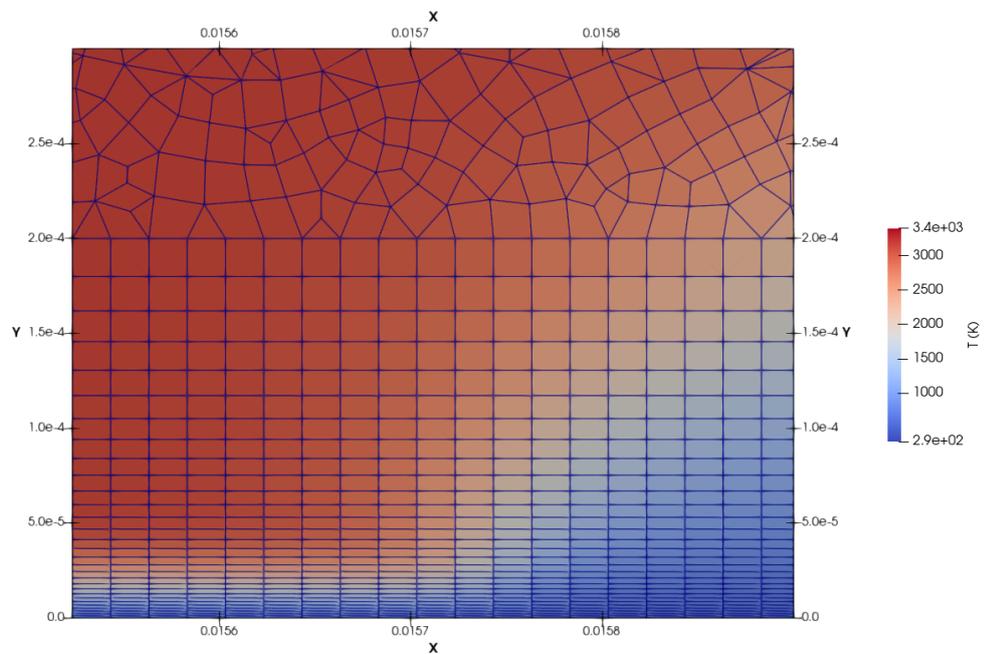
with  $y_1$  the height of the first cell above the wall,  $u_\tau$  the friction velocity and  $\nu = \mu/\rho$  the kinematic viscosity. The friction velocity is defined as:  $u_\tau = \sqrt{\tau_w/\rho_w}$  where the wall shear stress is  $\tau_w = \rho_w \nu (d_y u)_w$ . A large body of work has shown that, when  $y_1^+$  is low, typically lower than 5, the viscous stresses dominate and it can be shown that the velocity profile is of the form  $u \propto y^+$ . Hence, from a numerical point of view, the gradients within the boundary layer will be well approximated at the wall if the height of the first cell above the surface is slightly lower than  $y^+ = 5$ . A large



**Figure 11.3** Field of equivalent cell diameter for the fine mesh

amount of numerical work can be found in the literature, and the commonly accepted criteria for a proper spatial resolution of the boundary layer is  $y_1^+ \leq 1$ . Note that this criterion is especially suited to steady-state flows, however it remains sensible for unsteady applications.

*A posteriori* analysis of simulations with this mesh shows that the near-surface refinement with a first cell height of  $\delta = 1 \mu\text{m}$  yields a dimensionless first cell height  $y_1^+ < 0.2$  over the whole zone of interest, hence the boundary layer is correctly resolved in this RANS context. Figure 11.4 shows the mesh and the propellant flame near the ignition front for the detailed coupled approach. We can clearly observe the gradual mesh refinement near the surface, and the transition to the unstructured mesh for the rest of the domain.



**Figure 11.4** Details of the fine mesh near the propellant surface and ignition front

The propellant surface is discretised in 5097 boundary faces, each associated with an instance

name	$\delta$ ( $\mu\text{m}$ )	$\Delta x$ ( $\mu\text{m}$ )	number of CFD cells
fine	1	20	$3.48 \times 10^5$
intermediate	4	20	$2.92 \times 10^5$
coarse	20	20	$2.42 \times 10^5$

**Table 11.1** Main characteristics of the 3 meshes considered

of the one-dimensional model. The first 5000 boundary cells from  $x = 0$  m to  $x = 0.1$  m have a length  $\Delta x = 20$   $\mu\text{m}$ . From  $x = 0.1$  m to  $x = 2$  m, the mesh is gradually coarsened, such that only 97 boundary faces are added. The total number of cells in the CFD domain is  $3.48 \times 10^5$ . This mesh will be referred to as the fine mesh in the rest of this chapter. Each 1D propellant model has 60 cells for the solid phase, and 200 cells for the 1D flame (if activated). Note that a 20  $\mu\text{m}$  longitudinal refinement is on the order or even much smaller than the typical size of PA particles in usual AP-HTPB compositions (50 to 200  $\mu\text{m}$ ). However, the modelling approach chosen in Chapter 2 discards the geometrical representation of the heterogeneous nature of the propellant. Hence here, the longitudinal refinement is not required for geometrical reasons, but only for a proper spatial resolution of the coupled ignition model formulated in Chapter 7.

For the simulations with the 1D flame approach, issues with numerical stability of the current coupling have made it impossible to use the fine mesh. The root cause of this issue is discussed in Section 11.3.3. As a trade-off, we have been able to mitigate the instability by setting the first cell height to 4  $\mu\text{m}$ , which is still fine enough to resolve the near-surface flow dynamics. The lateral mesh resolution at the surface remains at 20  $\mu\text{m}$  and the unstructured part of the mesh is unchanged. This intermediate mesh has  $2.92 \times 10^5$  cells. This yields  $y_1^+ < 0.7$ . This mesh is referred to as intermediate.

To study the effect of the near-surface mesh resolution, another mesh was created, with the first cell height set to 20  $\mu\text{m}$ , all other parameters being unmodified. The resulting coarse mesh has  $2.42 \times 10^5$  cells and *a posteriori* analysis shows that  $y_1^+ < 4$ . The 3 meshes only differ by the vertical geometric progression inside the structured layer. They all have the same cell height and width at the top of the structured layer, hence the unstructured mesh is unaffected. Their main characteristics are summarised in Table 11.1.

#### 11.1.4 Numerical setup

The coupled integration of VULC1D and CHARME is performed using the first-order coupling algorithm described in Chapter 7. CHARME uses a linearised implicit Euler scheme solved with GMRES, while VULC1D uses the fully nonlinear implicit Euler scheme. The coupling time step is taken as the CFD time step from CHARME. Simulations are carried out either with a fixed time step, or with a dynamically adapted time step such that the relative solution variation in each cell is below 2%, following the procedure described in Section 7.2.3.

The detailed coupled approach is run on the fine mesh, while the 1D flame approach is run on both the intermediate and coarse meshes.

## 11.2 Reference result obtained with the detailed approach

A first simulation is run on the fine mesh with the detailed coupled approach to obtain a reference solution. The time step used is fixed and set to a sufficiently small value  $\Delta t = 2 \times 10^{-8}$  s, such

that the unsteady evolution is precisely resolved, even with the first-order temporal scheme. Thus, this simulation is termed as “reference” because all the flow field dynamics can be considered as properly resolved, both in terms of spatial and temporal discretisations. In this section, we analyse the evolution of the chamber flow field and propellant surface.

### 11.2.1 Dynamics of the combustion chamber

#### Global flow field

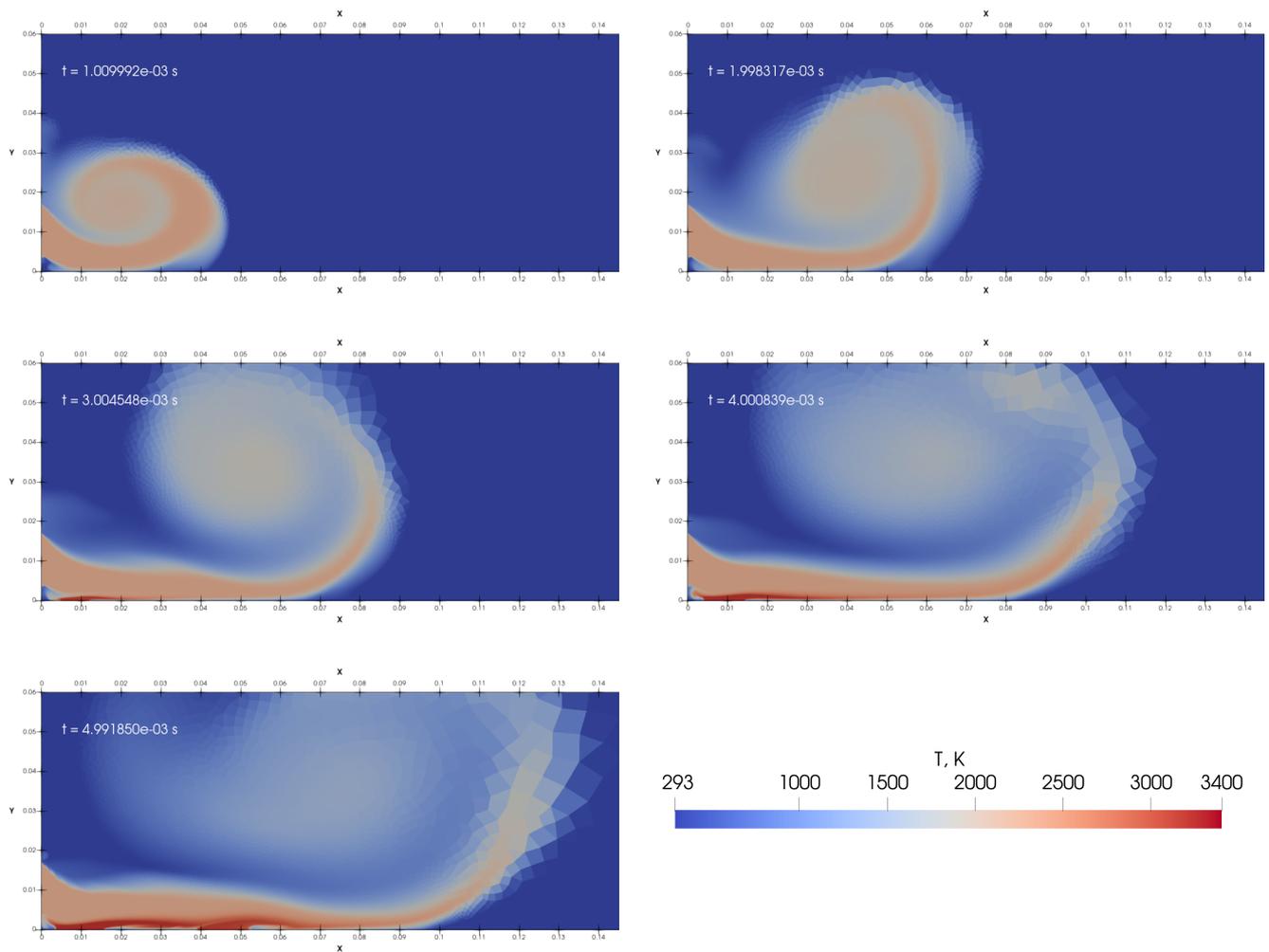
Various snapshots of the temperature flow field are shown in Figure 11.5 for the reference simulation. The initial impingement of the igniter jet is clearly visible, as well its progression above the propellant surface. From the third snapshot ( $t = 3$  ms) onwards, an ignited zone appears around  $x = 0.01$  m, and a second ignited zone appears in the last snapshot ( $t = 5$  ms) between  $x = 0.04$  m and  $x = 0.05$  m.

The first ignited zone produces a strong parietal injection which deflects the igniter jet away from the propellant surface, creating a recirculation zone ahead of the ignition front. This effect is also exacerbated by the two-dimensional configuration.

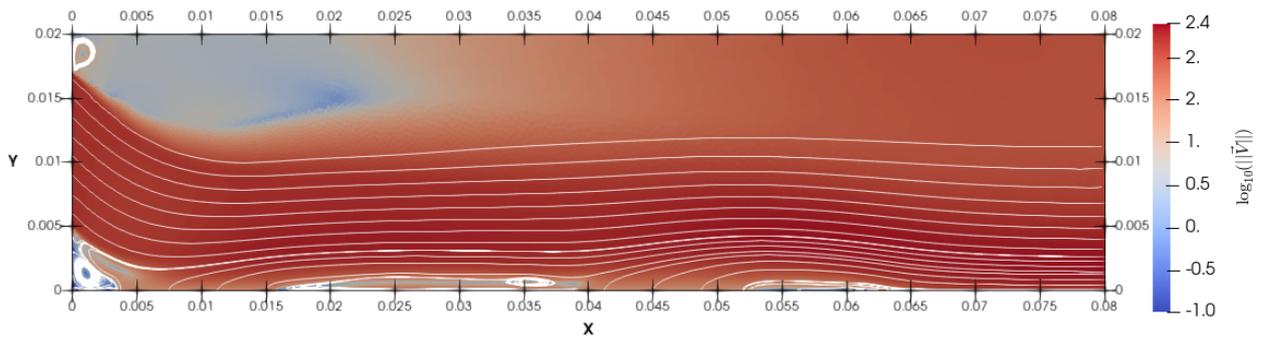
At  $t = 5$  ms, we can spot some irregularities of the upper boundary of the igniter jet, which would later transition to Kelvin-Helmholtz type instabilities. This self-destabilisation is more contained and appears later than the one observed in the 2D axisymmetric case from our article [70], where pressure waves reflected on the boundary conditions would trigger the instability sooner. This destabilisation was one of the root cause of the previously mentioned strong sensitivity of the initial test case used in that article.

#### Near-surface flow

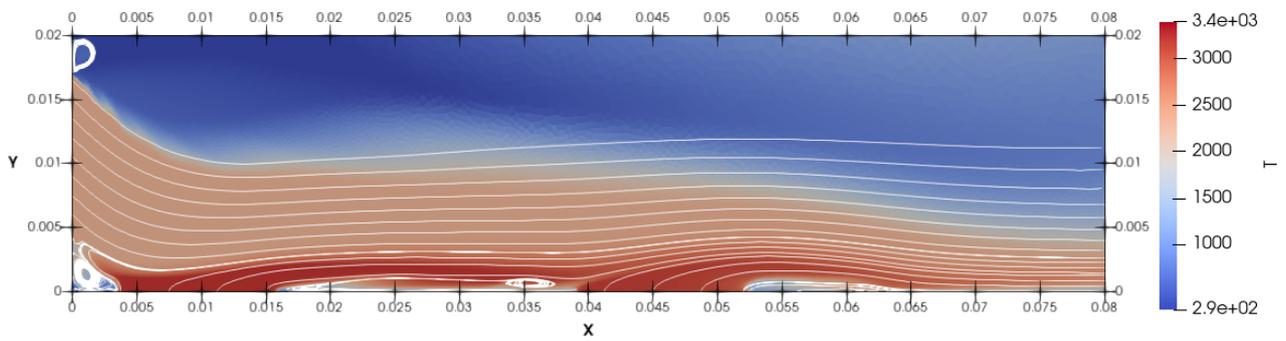
Let us focus on the zone of igniter impingement. Figure 11.6 shows the flow field near the ignited zones at  $t = 5$  ms for the reference simulation. The igniter jet flow and the flow coming from the ignited surfaces are clearly distinguishable, as seen from the temperature and flow direction discrepancies. At the ignited surface, the flow field coming from the surface pyrolysis is accelerated via thermal expansion in the propellant flame. Initially vertical, its direction quickly changes due to the interaction with the igniter jet. This flow turning is already slightly apparent in the flame zone, see Figure 11.9a. The flame is therefore slightly bent, which should in theory lead to an increased wall heat flux compared to the unbent case. The angle of the flow velocity vector with respect to the vertical is however very limited in the flame zone ( $< 0.1$  deg), therefore simple geometrical considerations show that the increase of the heat flux normal to the surface is on the order of 0.01% compared to the unbent case, hence is insignificant. The turbulent viscosity  $\mu_t$  has also been analysed in this region and is completely negligible in the whole flame zone. Therefore, no turbulent enhancement of the parietal heat transfer occurs with the detailed approach, as tacitly assumed in the 1D flame model.



**Figure 11.5** Evolution of the temperature field in the reference simulation



(a) Logarithm in base 10 of the velocity magnitude, with streamlines. Data is clipped to better highlight flow features.



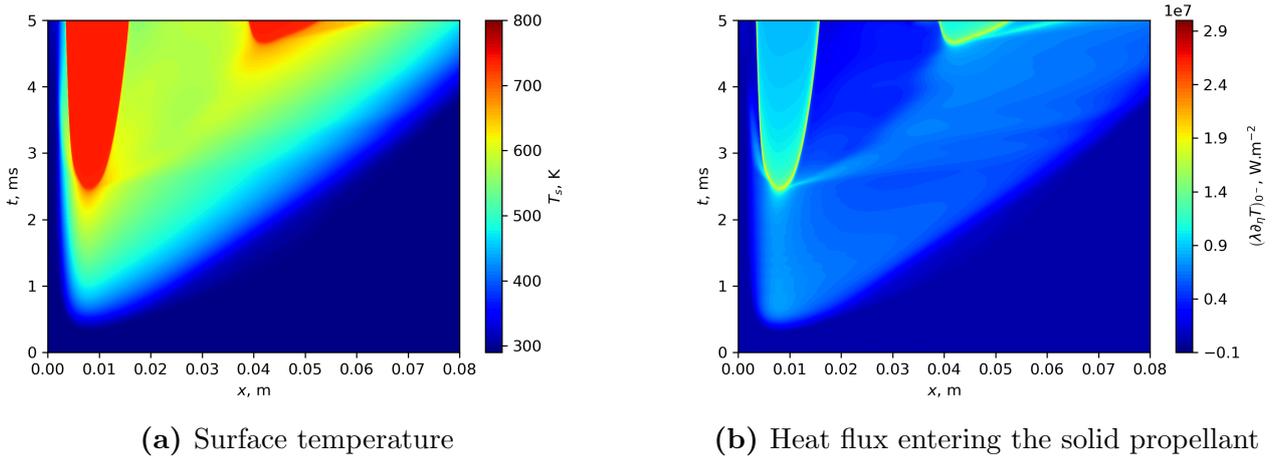
(b) Temperature field and streamlines

**Figure 11.6** Zoom on the flow field around the ignited zones at  $t = 5$  ms with the detailed approach on the fine mesh

## 11.2.2 Ignition dynamics

### Surface evolution

To better visualize the heating of the propellant surface and the propagation of ignition, Figure 11.7a shows the space-time diagram of the surface temperature for the reference simulation, while Figure 11.7b shows the space-time diagram of the heat flux  $(\lambda_c \partial_\eta T) (0^-)$  that enters the solid phase. These visualisations give a clear representation of the ignited zones and of the temporal evolution of the grain heating.



**Figure 11.7** Evolution of the propellant surface for the reference simulation

The igniter jet impinges the surface at  $t = 0.4$  ms and  $x = 7.8 \times 10^{-3}$  m, as indicated by the first surface temperature increase. The first ignition occurs at  $t = 2.4$  ms at the same location. Ignition then propagates around this point, thanks to the heat already accumulated in the solid propellant and to the increased heat flux caused by the propellant flame. The propellant combustion products that are expelled by the first ignited zone are quickly convected inside the lower part of the igniter jet, which leads to a temporary increase of the wall heat flux in the neighbourhood of the first ignited point, revealed by the thin, nearly horizontal, cyan ( $\Phi_w \approx 10$  MW.m<sup>-2</sup>) line in Figure 11.7b, starting from the point of first ignition.

The igniter jet is deflected by the pyrolysis mass flow rate from the first ignited zone, and thus a portion of the propellant surface is isolated from the convective heat transfer, as seen by the low-flux region near  $x = 25$  mm and  $t = 4$  ms.

## 11.3 Comparison of both modelling approaches

We now compare the previous reference simulation with the results obtained using the 1D flame approach.

### 11.3.1 Integration with a fine time step

A simulation with the latter approach is conducted on the intermediate mesh with the same time step  $\Delta t = 2 \times 10^{-8}$ s, and for the same physical time of 5 ms. To study the effect of the CFD mesh refinement, another simulation is run on the coarse mesh with the 1D flame approach and the same

settings. In this section, we compare the flow field evolution and the surface evolution with that of the reference simulation.

### Flow field evolution

The evolution of the temperature field obtained with the 1D flame approach is given in Figure 11.8b for the intermediate mesh and in Figure 11.8c for the coarse mesh.

On the intermediate mesh, we can observe that the overall evolution is very close to that of the reference simulation. In particular, up to 3 ms, the flow fields of both simulations are virtually identical. The first ignited zone has a very similar development in both cases. The only difference seems to be the second ignited zone, visible in the last snapshot, which is slightly smaller than in the reference simulation, and is also marginally more to the right of the domain. The discrepancy is however minor.

On the coarse mesh, the flow field is also very comparable up to ignition. At  $t = 3$  ms however, we see that there is already a larger volume of burnt gases, because ignition occurs slightly sooner, as we will discuss in the next section. At  $t = 5$  ms, the first ignited zone is very comparable to that of the reference simulation, however the second zone is slightly more extended to the left, causing a larger deflection of the igniter jet flow.

### Near-surface flow

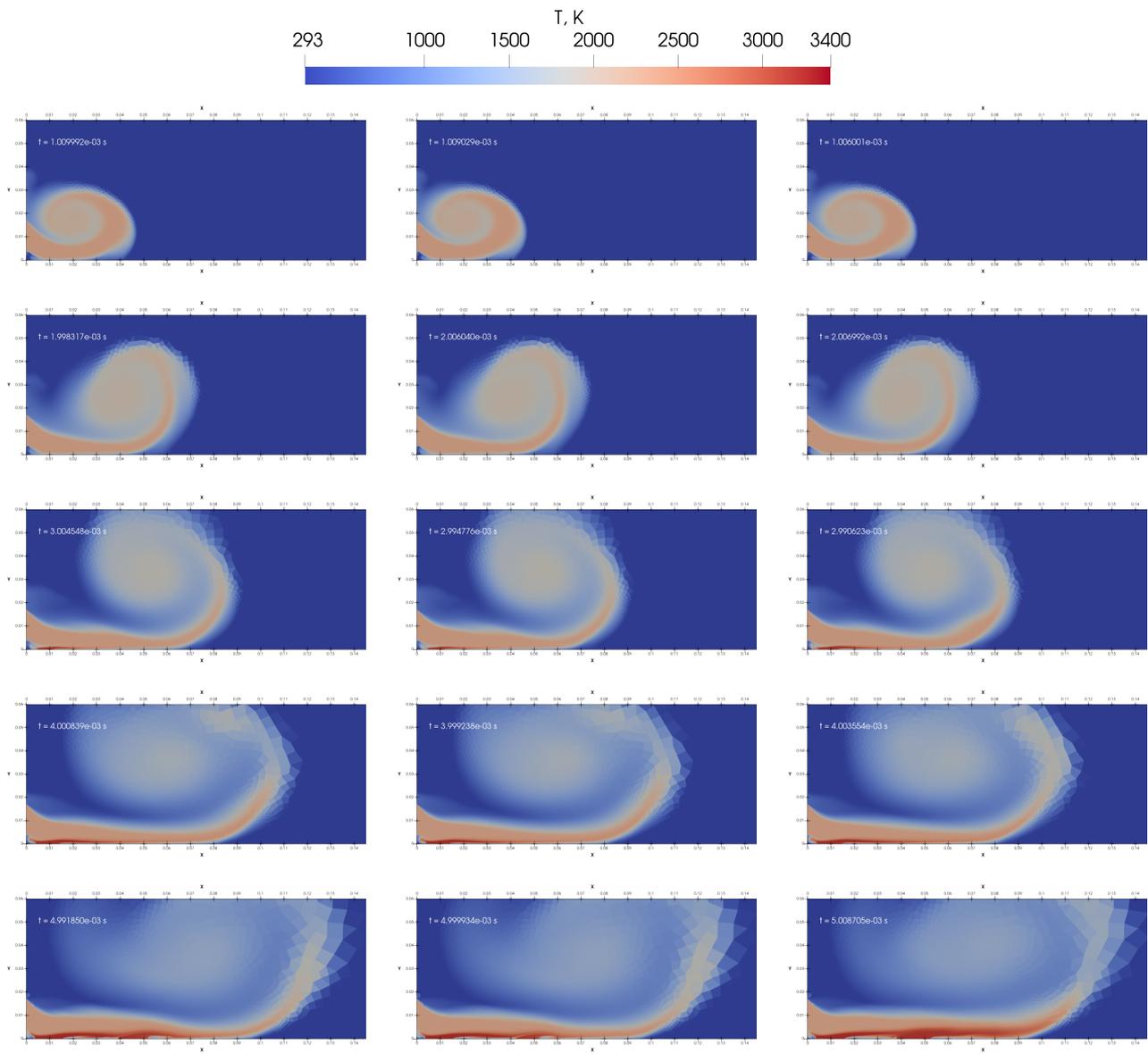
In Figure 11.9, the velocity field near the surface inside the first ignited zone at  $t = 5$  ms is plotted for these 2 simulations and the reference simulation. We see that the streamlines are very similar, and the velocity magnitudes in the upper layer of the boundary flow are in very good agreement. The only major difference is the effect of the thermal expansion in the case where the flame is solved within the CFD model. With the 1D flame approach, this thermal expansion is already accounted for in the 1D flame model and therefore is not visible in the CFD solution.

The similarity between the solution profiles is a very good *a posteriori* verification that the propellant flame can be concentrated in a boundary model instead of being solved within the CFD model, without much effect on the overall result.

Something to keep in mind is that the surface friction is different between both approaches, as the parietal flow speed is not the same in the 1D flame and detailed simulations. Indeed, both approaches inject in the CFD domain a similar mass flow rate, but not the same temperature, hence parietal densities are not the same: it is higher with the detailed approach than with the 1D flame approach. Using the ideal gas law (2.19), the ratio of both densities can indeed be obtained as the ratio  $T_f/T_s$ . As the mass flow rates are equal, the ratio of velocity is the inverse of that. This discrepancy is due to the fact the 1D flame model already accounts for thermal expansion, whereas this occurs within the flame in the CFD domain for the coupled approach. Consequently, the injection speed is roughly 4 times faster with the 1D coupled flame approach, and the friction forces differ. Further investigating this effect is out the scope of the present work, however analysis of the present simulations has shown that the surface friction forces are similar and have no particular influence on the overall flow field.

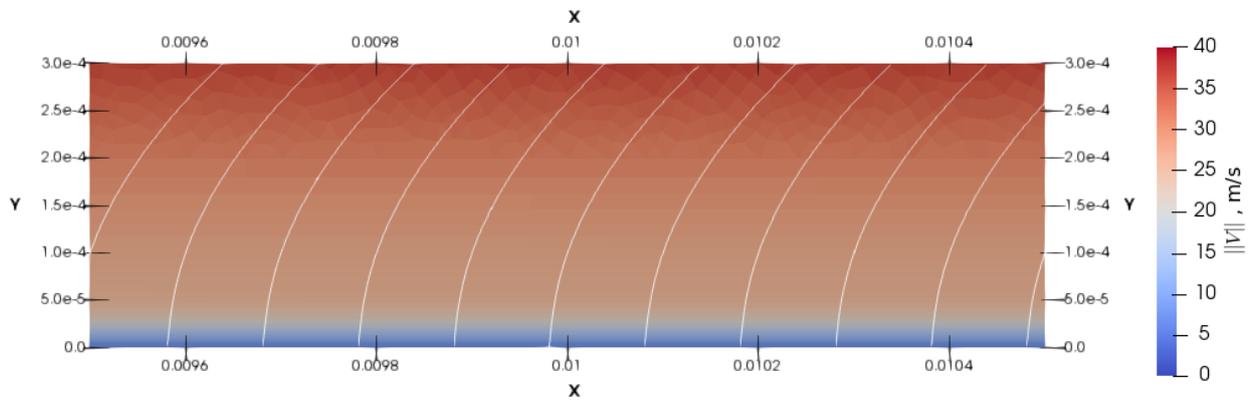
### Surface evolution

To better visualize the heating of the propellant surface and the propagation of ignition, Figures 11.10a, 11.11a and 11.12a show space-time diagrams of the evolution of the surface temperature

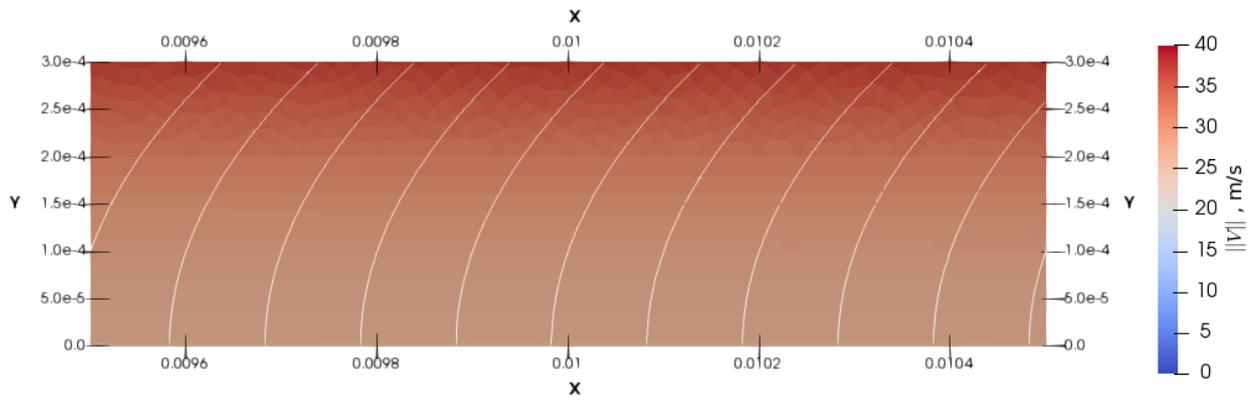


(a) Detailed model ( $\delta = 1 \mu\text{m}$ )      (b) 1D flame ( $\delta = 4 \mu\text{m}$ )      (c) 1D flame ( $\delta = 20 \mu\text{m}$ )

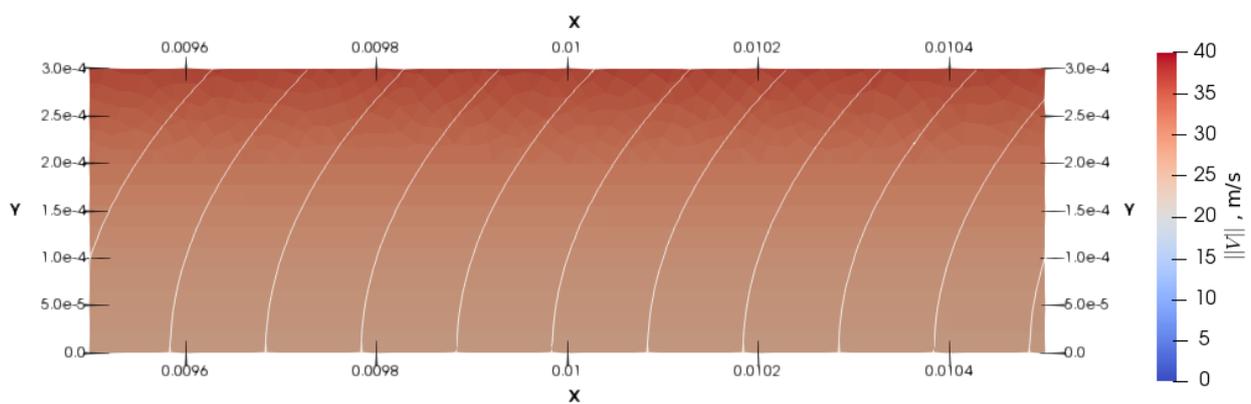
**Figure 11.8** Evolution of the temperature field with  $\Delta t = 2 \times 10^{-8}$  s. From left to right: detailed coupling, 1D flame coupling ( $\delta = 4 \mu\text{m}$ ), 1D flame coupling ( $\delta = 20 \mu\text{m}$ ). From top to bottom:  $t = 1, 2, 3, 4, 5$  ms.



(a) Detailed approach,  $\delta = 1 \mu\text{m}$



(b) 1D flame approach,  $\delta = 4 \mu\text{m}$



(c) 1D flame approach,  $\delta = 20 \mu\text{m}$

**Figure 11.9** Velocity field near  $x = 0.01 \text{ m}$  at  $t = 5 \text{ ms}$

for each simulation. Figures 11.10b, 11.11b, 11.12b show the space-time diagrams of the heat flux  $(\lambda_c \partial_\eta T)(0^-)$  that enters the solid phase. This allows for a clear representation of the ignited zones and of the temporal evolution of the grain heating.

First, we can observe that all results are overall in excellent agreement for the first 4 ms. In the case of the 1D flame approach with the intermediate mesh, ignition occurs at the same time and location ( $t = 2.4$  ms and  $x = 7.8 \times 10^{-3}$  m) as for the reference simulation. Note that the curious patterns in the heat flux for the 1D flame approach on the intermediate mesh are clear signs of the coupling instability discussed later in Section 11.3.3. For the 1D flame approach with the coarse mesh, ignition occurs slightly sooner at  $t = 2.15$  ms and at the same location, i.e. the near-surface mesh resolution has a strong impact on the occurrence of first ignition. We actually observe that, right from the start of the igniter jet impingement ( $t \approx 0.4$  ms), the heat flux  $(\lambda_c \partial_\eta T)(0^-)$  is larger in the simulation with  $\delta = 20$   $\mu\text{m}$ . This indicates that the conjugate heat-transfer is not well-resolved with the coarse mesh.

The first ignited zone produces a strong parietal injection which deflects the igniter jet away from the propellant surface. This creates a recirculation zone ahead of the ignition front, which also increases the overall sensitivity of the surface ignition, because the recirculation zone is very little exposed to the convective heat transfer from the igniter flow, as is clearly seen by the very low heat flux values in the previous space-time diagrams of  $(\lambda_c \partial_\eta T)(0^-)$ .

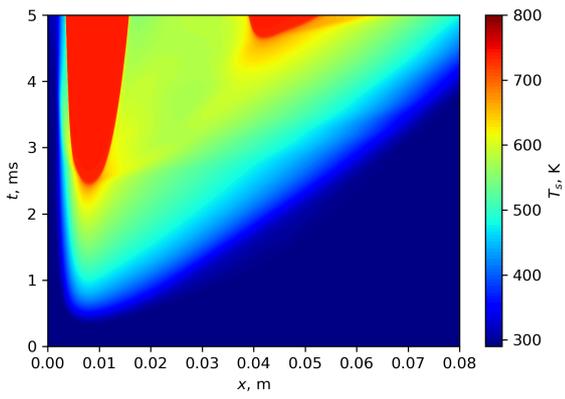
A second ignition zone appears near  $t = 5$  ms and  $x = 5 \times 10^{-2}$  m in all simulations. The detailed model on the fine mesh and the 1D flame model on the intermediate mesh have a very similar development for this zone, although ignition occurs slightly sooner with the detailed model. On the coarse mesh however, ignition of the second zone occurs much sooner. This can be once more attributed to the poorer resolution of the conjugate heat-transfer. Indeed, we see that the heat flux that enters the solid phase is always higher at any point of the surface during the initial heating on the coarse mesh, compared to the other configurations. Also, since the first zone ignites earlier, the jet deflection is not the same, and the point where the igniter jet once again impinges the surface (having incorporated some hotter gases from the propellant combustion in the first zone) is different. Consequently, the heating of the downstream unignited regions is modified and the secondary ignition zone appears at slightly different times and locations. This effect is also exacerbated by the two-dimensional configuration, because the recirculation zone cannot be circumvented.

Overall, we see that the 1D flame approach yields quantitatively very similar results compared to the detailed approach if the CFD mesh is sufficiently refined. If a coarse mesh is used, the discrepancies can be attributed to a poorer resolution of the chamber fluid model hydrodynamics and conjugate heat transfer.

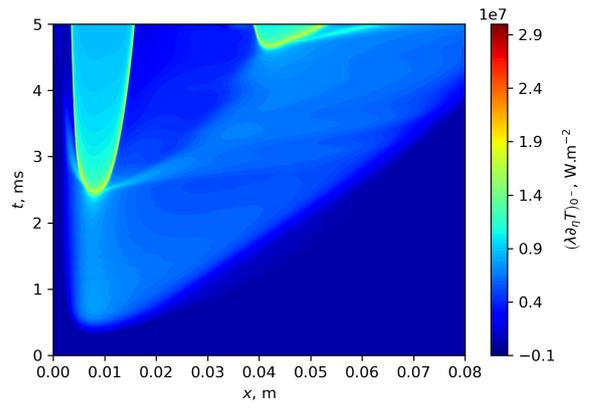
### 11.3.2 Influence of the temporal resolution

The previous simulations were conducted with a fixed time step  $\Delta t = 2 \times 10^{-8}$  s which is quite low for practical applications. Therefore, it is interesting to see how the overall ignition dynamics is affected when larger time steps are used. To this end, the 3 simulations are run anew with a dynamic time step, following the basic time step adaptation method presented in Section 7.2.3. The maximum relative solution variation from one step to the next is set to 2%.

The space-time diagrams of  $T_s$  and  $(\lambda_c \partial_\eta T)(0^-)$  are presented in Figures 11.13a, 11.14a and 11.15a. The increased time step has a slight impact on the flow field. This can for example be noticed by observing that there is a slight variation in the speed of propagation of the igniter

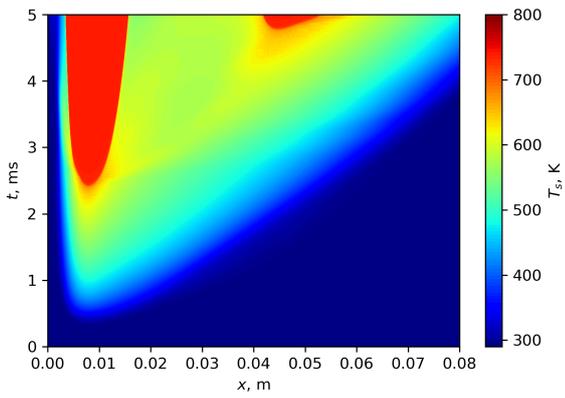


(a)  $T_s$

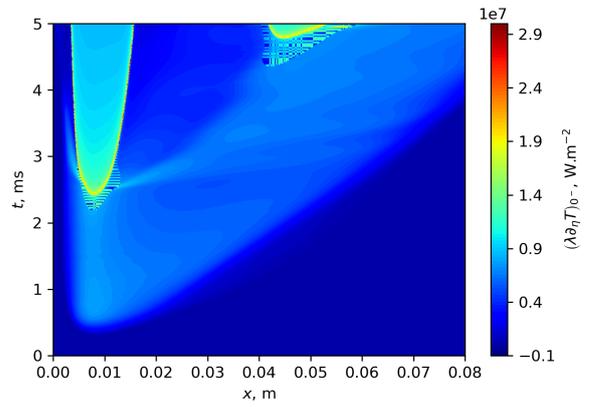


(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.10 Detailed approach ( $\delta = 1 \mu\text{m}$ ) with  $\Delta t = 2 \times 10^{-8} \text{ s}$

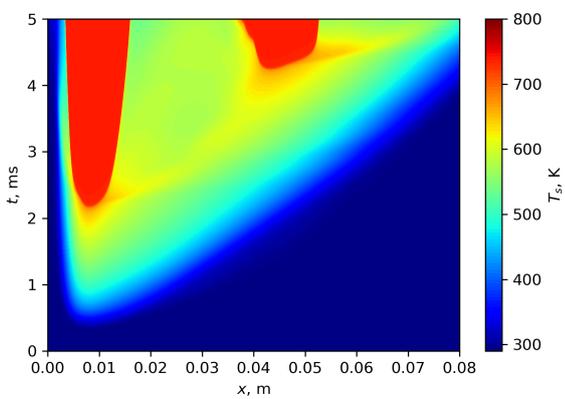


(a)  $T_s$

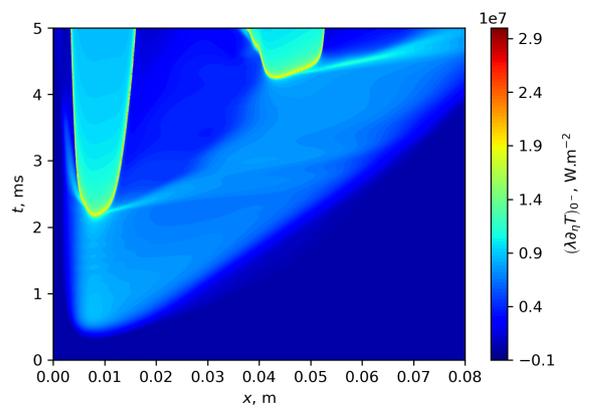


(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.11 1D flame coupled approach ( $\delta = 4 \mu\text{m}$ ) with  $\Delta t = 2 \times 10^{-8} \text{ s}$



(a)  $T_s$



(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.12 1D flame coupled approach ( $\delta = 20 \mu\text{m}$ ) with  $\Delta t = 2 \times 10^{-8} \text{ s}$

jet along the surface. Indeed, for  $\delta = 1 \mu\text{m}$  and  $\delta = 4 \mu\text{m}$ , the propagation is slightly faster with  $\Delta t = 2 \times 10^{-8} \text{ s}$ , as can be seen by comparing the temperature fields, e.g. Figures 11.10a and 11.13a. Most likely, this is due to a difference in the temporal resolution of the near-surface hydrodynamics within the fine structured layer of the mesh, in particular due to the use of a first-order time scheme. Still, these differences are quite small, hence they are not investigated further.

The temporal evolution of the heat flux at the position of first ignition is plotted in Figure 11.16, while the relative difference to the heat flux from the reference simulation (detailed model with the finest time step) is plotted in Figure 11.17. It should be kept in mind that this relative comparison does not make sense during the start of the simulation, where the heat flux is essentially 0, and during the ignition phases ( $t \approx 2.2 \text{ ms}$ ), as these are not exactly simultaneous.

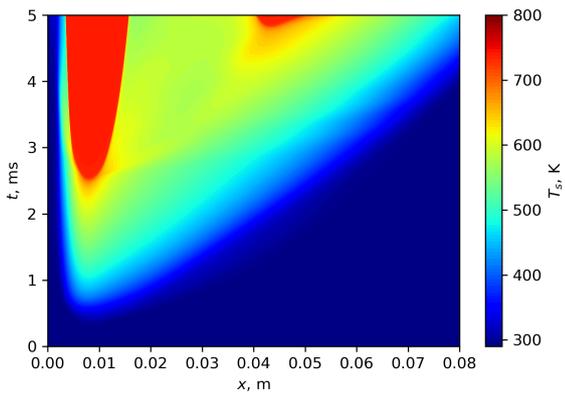
A first observation is that, during most of the initial inert heating phase, the heat flux entering the solid phase in the simulations with  $\delta = 20 \mu\text{m}$  is roughly 10 to 12 % higher than in the other simulations, independently the time step used. This indicates that the conjugate heat transfer at the surface is not well resolved, and that a mesh refinement is mandatory to obtain a truly converged initial heating. This is coherent with the fact that  $y_1^+$  is higher than 1, whereas  $y_1^+ < 1$  is a well-know requirement for a proper spatial resolution of the boundary layer in the RANS context.

On the opposite, we see that for the simulations on the fine and intermediate meshes, the heat fluxes are always within 2 % of each other during most of the time range considered. This shows that the conjugate heat transfer is well resolved, as opposed to what we have just commented on the coarse mesh. The transition to ignition is clearly visible by the spike in heat flux. These spikes are almost identical between both simulations with the intermediate mesh, and are very close to the spike from the reference simulation with the detailed approach. This indicates that the 1D flame approach enables an accurate capture of the grain heating, and that the necessary transition of its wall heat flux formulation during ignition, as discussed in Section 7.3, is correct, or at least does not have an important effect on the overall dynamics. This is further verified by observing that the heat flux spikes at ignition on the intermediate and coarse meshes are of the same height, i.e. the effect of the first cell height  $\delta$  on the  $\Phi_w$  is minimal.

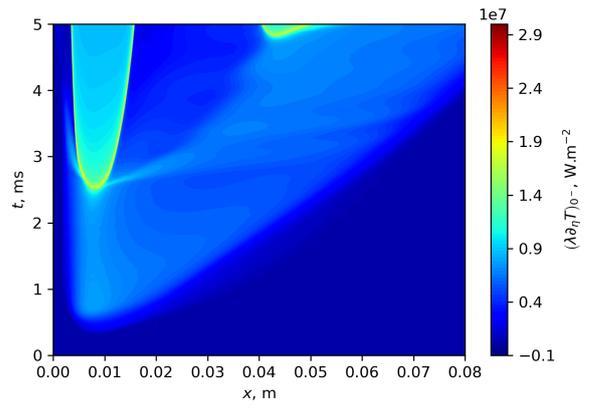
The result obtained with the detailed approach on the fine mesh and a dynamic time step is quite different from the reference simulation. Already in the inert heating phase, we see that there is a delay of approximately 0.1 ms for the initial build-up of the heat flux in Figure 11.16. The ignition spike also suffers from a similar delay. In fact, a careful comparison of Figures 11.10a and 11.13a shows that the whole inert heating is delayed in the same manner. This surprising effect of the time step reduction can only be traced back to the first-order accuracy and the fine spatial mesh, which lead to an overall poorer temporal resolution of the dynamics.

Another interesting observation can be made by looking at the post-ignition phase. All the fluxes seem to converge to a steady-state that is reassuringly independent of the time step used. Looking at the ratios of these heat fluxes, we see that all simulations yield similar heat fluxes within 1 %. In particular, the simulations with the 1D flame approach converge to a steady-state heat flux which is approximately 1 % higher than the value obtained with the detailed model. This translates into a higher pyrolysis mass flow rate (roughly 2 %) at the surface compared to the detailed approach. This is the same phenomenon as observed for the purely one-dimensional combustion in Chapter 6. The difference is not linked to the spatial resolution (or lack thereof) of the propellant flame. The origin of that discrepancy has no yet been found. Still, the difference remains very acceptable.

The time steps used during the simulations are plotted in Figure 11.18 for the simulations where the constraint  $\Delta t \leq 2 \times 10^{-6} \text{ s}$  is applied. The red curve corresponds to the simulation with the finest mesh and the detailed coupled approach. We can see that the time step is, in average, reasonably

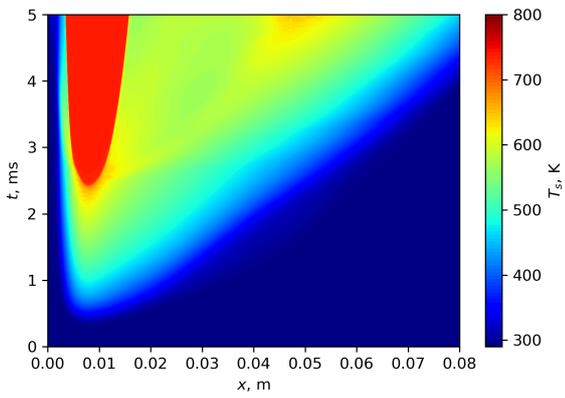


(a)  $T_s$

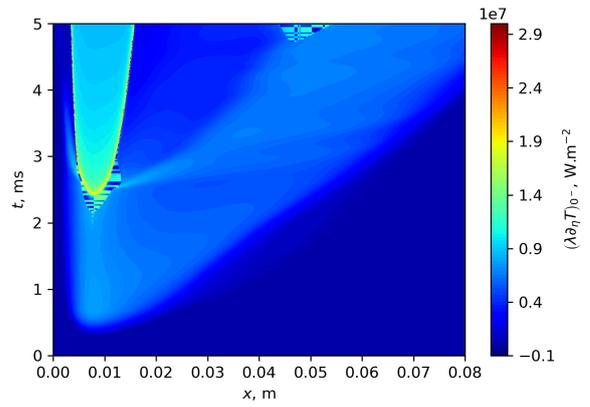


(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.13 Detailed approach ( $\delta = 1 \mu\text{m}$ ) with  $\Delta t \leq 2 \times 10^{-6} \text{ s}$

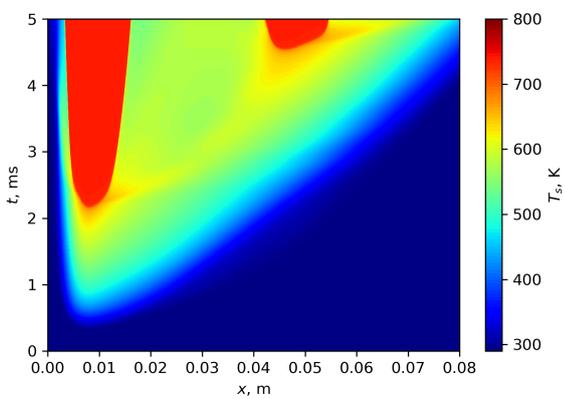


(a)  $T_s$

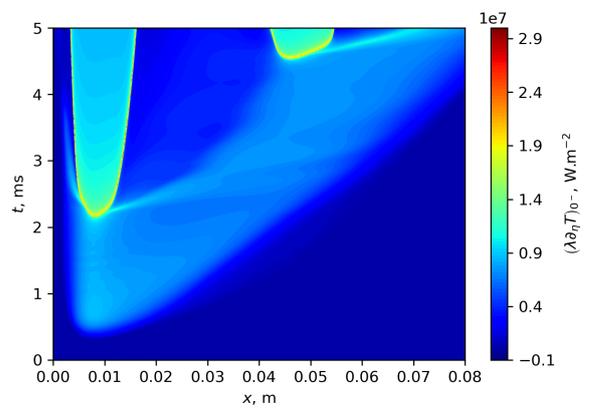


(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.14 1D flame coupled approach ( $\delta = 4 \mu\text{m}$ ) with  $\Delta t \leq 2 \times 10^{-6} \text{ s}$

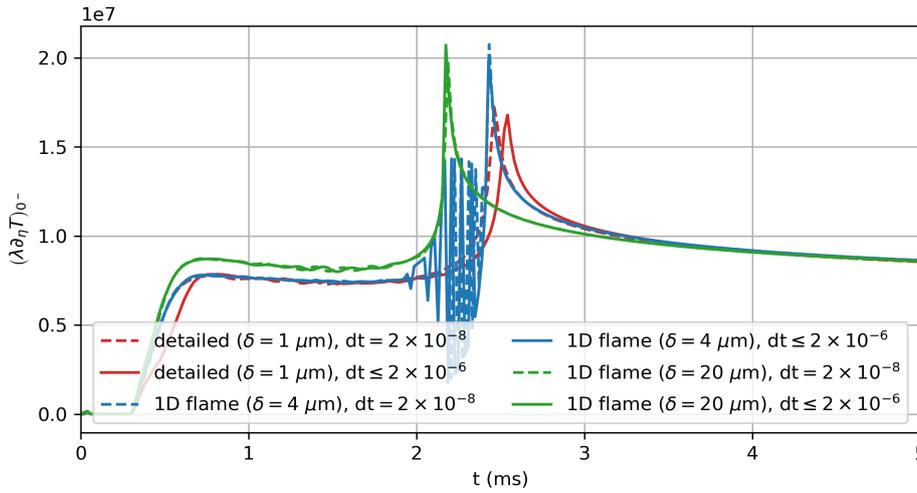


(a)  $T_s$

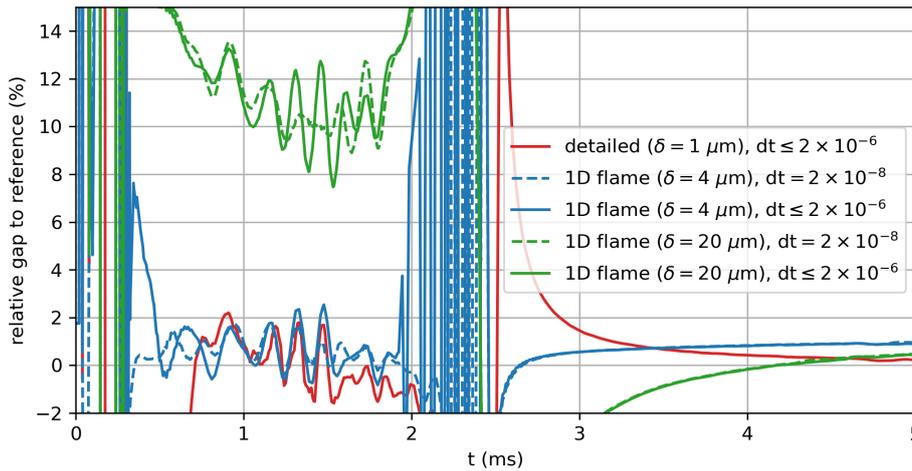


(b)  $(\lambda_c \partial_\eta T) (0^-)$

Figure 11.15 1D flame coupled approach ( $\delta = 20 \mu\text{m}$ ) with  $\Delta t \leq 2 \times 10^{-6} \text{ s}$

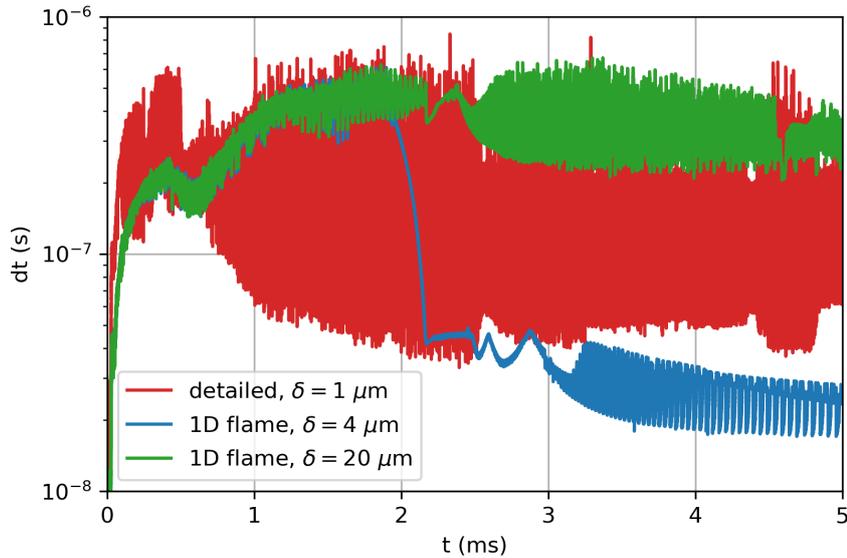


**Figure 11.16** Heat flux entering the solid phase at  $x = 7.8 \times 10^{-3}$  m



**Figure 11.17** Relative difference of the solid phase heat flux compared to the reference simulation at  $x = 7.8 \times 10^{-3}$  m

high. However, the simulations occasionally suffer from strong variations. This phenomenon is quite striking in the case of the detailed simulation on the finer mesh. Lowering the maximum allowed time step increase may be a first solution. Also, a time step controller could be implemented, similarly as done for embedded and multistep integration methods [231] to smooth the time step evolution. The time step control strategy used here is based on a very pragmatic approach, and by no means ensures a given error level on the overall temporal evolution, as opposed to time adaptive schemes as used in Chapter 5. Potential solutions to improve that aspect have been discussed in Chapter 8. Nonetheless, this strategy has allowed to obtain overall stable solutions, avoiding numerical convergence issues we often encountered with a fixed time step and highly transient dynamics, in particular due to the coupling instability discussed in Section 11.3.3. Note that, although not shown here, simulation with the intermediate mesh and the detailed model did not suffer from such a large time step variability. This indicates that the instability is caused by the near-surface mesh



**Figure 11.18** Evolution of the CFD time step

refinement. Indeed, the Neumann conditions imposed to the CFD solver CHARME are imposed explicitly, and this is known to cause numerical instabilities when using fine meshes [256].

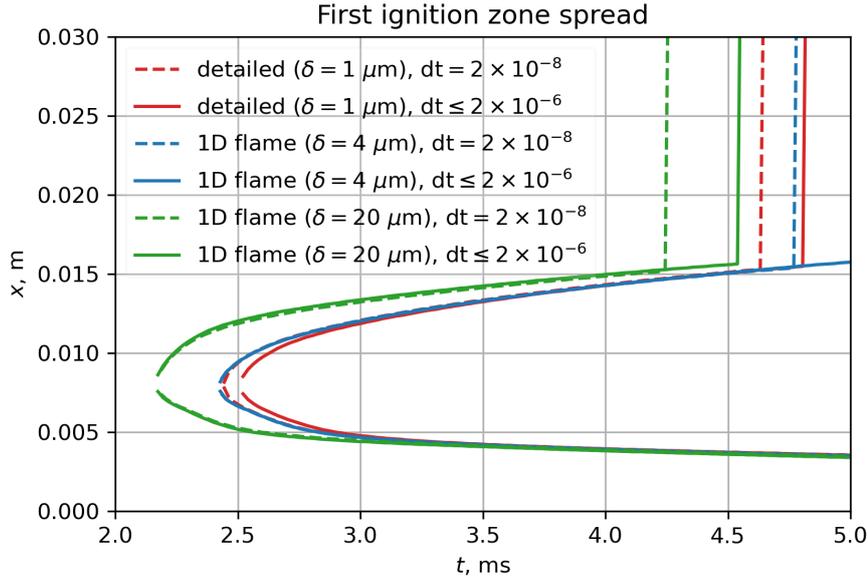
For the simulation with the 1D flame approach and the intermediate mesh (blue curve), we can clearly observe the onset of the surface coupling instability near  $t = 2$  ms, which causes an order of magnitude decrease in the time step values used, approximately by a factor 10, almost reaching the low time step used for the previous simulations with fixed time step. Note however that, until ignition, the time step is on the order of  $2 \times 10^{-7}$  s or higher, without a noticeable impact on the solution dynamics. This demonstrates that the computational cost can be greatly reduced by increasing the time step, without sacrificing much on accuracy. The lowering of the time step is linked to a coupling instability which is discussed in more details in Section 11.3.3 and is solely linked to the transition to ignition of each individual 1D model used in the simulation.

The simulation with the 1D flame approach on the coarse mesh initially follows the same time step evolution, however it does not suffer from that time step reduction at ignition, and the time step level remains in average equivalent before and after the first ignition.

Figure 11.19 shows the temporal evolution of the leftmost and rightmost ignited abscissa along the propellant surface. Again, we clearly see the excellent agreement between the 1D flame approach on a fine mesh ( $\delta = 4 \mu\text{m}$ ) and the detailed coupled approach ( $\delta = 1 \mu\text{m}$ ) if the latter is run with the finest time step. With the third mesh ( $\delta = 20 \mu\text{m}$ ), the first ignition occurs slightly sooner at the same point, and the flame propagation along the surface is slightly faster, which has a visible impact on the dynamics of the chamber flow field.

### 11.3.3 Investigation of the instability for the 1D flame approach

In the result obtained with the 1D flame approach and the intermediate mesh ( $\delta = 4 \mu\text{m}$ ), we can spot a slight surface temperature instability before the first ignition, as can be seen in Figure 11.14 with the orange and yellow stripes in the temperature field near  $x = 8 \times 10^{-3}$  m and  $t = 2.4$  ms. It is worth noting that this instability also appears with  $\Delta t = 2 \times 10^{-8}$  s, but only later for instance



**Figure 11.19** Temporal evolution of the outermost ignited positions

at the location of first ignition.

Originally, we had run the 1D flame approach with the finer mesh ( $\delta = 1 \mu\text{m}$ ) and the computation would suffer from a much more violent instability near the first ignition point, where the wall heat flux would steadily increase in norm, but its sign would change at each time step. The dynamic lowering of the time step was not sufficient to dampen the instability and numerical convergence errors would ensue, making it impossible to drive the simulation past the first ignition.

We first thought that this instability was due to the lack of implicitness of the wall heat flux  $\Phi_w$  in VULC1D. Indeed the formulation of that flux involving  $T_f$  breaks the tridiagonality of the Jacobian of the 1D model, because the wall heat flux is imposed at the propellant surface and depends on  $T_f$  which is the temperature of the last 1D gas cell, and not that of the surface or of a cell adjacent to the surface. Hence the 3-point stencil from the 1D spatial discretisation is not respected any more. Recovering the tridiagonality could be done by adding a dummy field  $\Lambda$  with  $d_x \Lambda = 0$  and  $\Lambda(+\infty) = T_f$ , however issues with the VULC1D code did not permit to successfully implement this improvement. Hence,  $\Phi_w$  could only be handled explicitly for VULC1D, i.e. it is computed once at the start of the VULC1D step and is an imposed constant at each step, instead of implicitly depending on the iteratively evolving 1D field variables such as the flame temperature.

The alternative wall heat flux formulation  $\Phi_w = h(T_1 - T_s)$  can however be implicitized without such issues. Experiments on the intermediate mesh with this formula have shown that the instability does not appear before ignition. Implicit and explicit handlings of  $\Phi_w$  yielded virtually identical results. Therefore, it seems the explicit treatment is not to blame for the instability originally encountered, but rather  $T_f$ .

In fact,  $T_f$  acts as the parietal temperature for CHARME, whereas  $T_s$  would play this role in a conventional conjugate heat transfer case. During the initial heat-up,  $m = 0$  hence a simple energy balance on the 1D quasi-steady gas flame yields  $T_f = T_s$ . The difference only appears once the quasi-steady propellant flame establishes itself. Then,  $T_f$  becomes larger than  $T_s$ , and we have,

locally at time  $t_0$ , a relationship of the form:

$$T_f(t) = T_f(t_0) + (\partial_{T_s} T_f)(t_0) (T_s(t) - T_s(t_0)) \quad (11.2)$$

We observed that, when the instability was triggered,  $\partial_{T_s} T_f$  was approximately equal to 3. Hence, the quasi-steady 1D flame acts as a multiplier for the surface temperature. This greatly decreases the stability of the conjugate heat-transfer on the CFD side, because a small variation on  $T_1$  may have a great impact on  $T_w = T_f$ .

Coarsening the CFD mesh near the surface is an efficient way to eliminate this instability, however at the cost of a poorer resolution of the surface heat transfer. Therefore, in all ignition simulations performed at ONERA, this instability does not appear, because near-surface mesh sizes are much larger to decrease the computational cost. Wall laws are used to improve the accuracy of the wall heat transfer, mitigating the impact of the poorer spatial resolution. The effect of such laws is however a complex matter in itself [245], therefore we do not investigate this further in the present work.

## Partial conclusion

The new strategy for the simulation of solid rocket motors at large-scale presented in Chapter 7 has been employed for a 2D test case, which has been designed to feature the various phenomena encountered in SRMs: igniter jet impingement, complex hydrodynamics, flame spread. The simulations presented in this chapter rely on all the models and code developments presented in the previous chapters.

The newly designed test case has proven to be highly sensitive, in particular due to the stringent near-surface mesh refinement required. We have seen that the criterion  $y_1^+ \leq 1$  must be respected in this RANS framework for a proper resolution of the conjugate heat transfer between the igniter flow and the propellant surface. Also, the two-dimensional setting allows for sustained recirculation zones to be created, isolating part of the unignited propellant surface from the igniter jet, and this effect is dependent on the propellant ignition, flame spread and mass flow rate distribution.

To our knowledge, this chapter, together with our article [70], are the first reports on the use of a detailed approach for the simulation of ignition. This modelling enables a more complete picture of the flame establishment and its interaction with the surrounding flow, compared to the 1D flame approach.

We have shown that the 1D flame approach and the detailed approach yield nearly identical results on fine meshes. This clearly demonstrates the accuracy of the 1D flame approach, even in highly-transient and multidimensional simulations. This *a posteriori* legitimates the use of such a modelling as in all simulations reported in the literature.

Compared to the existing ignition models, the specificity of our 1D flame approach lies in the numerical resolution of the propellant flame, which allows for a nearly arbitrary level of modelling to be employed, without resorting to very simplified flame models. Also, the transition to ignition occurs dynamically, which may lead to better fidelity across a wider variety of conditions than a simpler temperature criterion.

Coarsening the CFD mesh has a noticeable impact on the conjugate heat transfer, causing an overall faster heating of the propellant. However, this discrepancy could potentially be corrected by using wall laws to compensate for the poor near-surface resolution of the thermal boundary layer. That would greatly increase the advantage of the 1D flame model, because much coarser

meshes could then be used. Overall, this configuration gives a strong demonstration that the 1D flame model behaves correctly, and that it is a sensible modelling approach for large-scale ignition transient simulations.

In the present work, no wall laws have been used to improve the accuracy of the conjugate heat transfer. Future work should investigate the use of these, and the detailed coupling could be used to generate new wall laws better suited for a given configuration.

Further studies with various lateral and vertical near-surface mesh refinements should be conducted to assess the effect of spatial resolution on the flame spread. Radiation models available in CEDRE can easily be added to the overall modelling to account for gas and surface radiation. This will allow for radiative grain preheating to occur, which may soften the differences observed in the 2D test case, as radiation has often been reported to be an important contribution for the propagation of ignition [52].

Some aspects remain difficult to model with the 1D flame approach. For instance, at the aft-end of an SRM, turbulence may impact the flame structure [307] and thus the burning rate of the propellant. Our current 1D flame model does not account for any turbulence related effects. The inclusion of turbulence-related conservation equations in the 1D flame model could improve this aspect by providing an unsteady numerical wall law.

Finally, the presented coupling strategy is only first-order accurate in time. Both the CFD and propellant codes are however able to achieve higher-order separately. Hence future work should focus on implementing the adaptive coupling framework explored in Chapter 8 to improve the efficiency and accuracy of the coupling.

# Conclusions and perspectives



## Conclusion

In this work, we have presented several contributions towards more accurate simulations of SRM ignition. We have seen that such simulations are only viable if the propellant ignition and combustion is accounted for in a dynamic boundary condition for the CFD code that simulates the SRM internal flow field. This boundary condition can conveniently be a 1D model, which computes the unsteady heating of the propellant and its combustion. However, existing works on this topic use very simplified 1D models, which may limit the ability to reproduce the ignition behaviour of different kinds of propellant and the ignition dynamics of various motors. Therefore, we have chosen to develop a new 1D model with a higher level of physical refinement, which is much closer to detailed 1D models used for the study of detailed kinetics for instance.

This has been the focus of Part I. The model has been derived based on commonly accepted assumptions, and has been designed such that additional physics can be included easily in the future. To provide a strong mathematical basis for the model and to study its nature, a first analysis has been conducted in steady-state with some additional simplifications, leading to the formulation of a nonlinear eigenvalue problem, where the eigenvalue is the propellant surface regression speed. The existence and uniqueness of the solution has been proven, and the mathematical reasoning used for the proof can be directly translated into a numerical shooting method which can compute the solution profile to machine accuracy. This semi-analytical tool has been very helpful in generating reference solutions for the verification of other codes, as well as for performing extensive parametric studies efficiently.

To simulate the ignition and unsteady combustion of the propellant in 1D, the original model has been semi-discretised in space using a finite-volume approach. The resulting system of equations has been shown to be differential-algebraic of index 1, a property which had not clearly been identified in the community. Based on this observation, specific temporal integration schemes are required to obtain an accurate solution. The family of stiffly accurate L-stable ESDIRK methods with an embedded error estimate has been selected and adapted to the framework of 1D low-Mach combustion, with an original and natural treatment of the mass conservation constraint, which facilitates the convergence of the Newton iterations. Numerical results have shown that this enables to reach up to a fifth-order accuracy in time on all variables, which is a large improvement over methods traditionally used in the one-dimensional combustion community. The numerical strategy and the 1D model have been implemented in the sequential Fortran code `VULC1D`, and various test cases have shown the adequacy and performance of our numerical strategy, in particular for strongly nonlinear dynamics (ignition, instability leading to a limit cycle). The code is also versatile and robust, as it can handle complex models including detailed kinetics, as well as very fine spatial meshes efficiently. Time adaptation has been shown to be very valuable, as solutions can be obtained quickly with a guaranteed accuracy, allowing for a speed-up factor of 60 in restitution time for some applications over lower-order methods that do not rely on an error estimate.

To reproduce the ignition characteristics of various propellants, a constrained optimisation procedure has been developed, which allows to fit the model parameters so as to match experimental observations, in particular laser-induced ignition times. This approach has been shown to be robust and efficient, and the optimised parameter set closely reproduce the experimental data.

Following the development of the 1D code `VULC1D`, the coupling with the semi-industrial CFD toolchain `CEDRE` from ONERA has been presented to enable the simulation of ignition in a complete motor. The 1D model acts as a dynamic boundary condition which encapsulates the physico-chemical complexity associated with the surface regression and the propellant flame. The physical

fluxes exchanged by both solvers have been described, and a first-order staggered serial temporal coupling algorithm has been presented. Issues arising from the compression of the flame into a surface phenomenon from the viewpoint of the CFD solver have been discussed. In particular, ensuring the coupling is conservative and physically coherent requires the 1D flame to be quasi-steady. Also, the conjugate heat transfer at the propellant surface must be handled specifically, in particular when the surface of the propellant ignites. A variant of the 1D flame coupled approach has been designed, where the flame is solved in the CFD solver itself instead of VULC1D. This detailed coupling allows for a direct numerical simulation approach (not in the turbulence-related sense) of the propellant flame and its interactions with the chamber flow, within the simplified physico-chemical framework used in this work. A comparison of both approaches for laser-induced ignition in 1D has shown that they reproduce with a satisfying accuracy the reference result obtained with VULC1D, thus verifying the implementation of the coupling.

Strategies for increasing the order of the coupling and dynamically adapting the coupling time step have been discussed, and a demonstrator of an adaptive-order, adaptive-time-step co-simulation algorithm has been implemented in Fortran to couple a 0D chamber model with VULC1D. It shows interesting potential for application to the coupling with CEDRE and for other problems involving the coupling of various models across a physical interface, each model being simulated by a specific solver.

Numerous code developments have been performed during the course of this PhD thesis. Python has been used extensively for the implementation of prototypes and for the analysis of simulation results. The production code VULC1D and the coupling procedures with CEDRE have been written in Fortran. A Python interface to VULC1D has been developed, allowing for automated parametric studies and Python-based optimisation of the various model parameters.

The last chapter of this thesis has presented the convergence point of all of the above work: 1D model development, optimisation of the model parameters, coupling with CEDRE and the various code developments. A 2D test case, amenable to fully-resolved simulations with the detailed coupling, has been designed. It is representative of the impingement region of an igniter flow on a solid propellant surface in a small SRM, and includes many complex phenomena found in SRMs, making it a demanding and discriminating test case. Simulation of ignition with the 1D flame and the detailed approaches have shown that the 1D representation of the flame and its compression into a boundary phenomenon for the CFD solver yields almost identical results to the fully-resolved detailed approach. This is an important finding, as it legitimates the widespread use of a 1D propellant combustion model as a 1D dynamic boundary condition in a CFD solver. This 1D flame approach paves the way to more realistic 3D simulations of SRM ignition.

# Perspectives

The work presented in this manuscript is interdisciplinary and therefore leads to a number of perspectives for each discipline separately, but also for interactions between them.

## Theoretical analysis

The theoretical analysis of the steady travelling combustion wave can be extended. **Including a non-unity or solution-dependent Lewis number** in the gas phase should be considered, since it can lead to more realistic solution profiles. However, the existence and uniqueness of the solution may not be guaranteed any more [64]. This is also the case if more physics is included, e.g. surface thermal radiation.

The **stability of the steady-state solution profile** should be investigated, using advanced theoretical tools for dynamical systems based on a spectrum analysis of the continuous system [308]. Stability studies have already been presented for generic travelling wave problems in one-dimension involving a non-reactive interface condition [309, 310], or in the solid propellant case for a multidimensional model, but with a high-activation energy assumption for the gas flame [311].

The **effect of gas-phase unsteadiness** on the propellant response to pressure fluctuations has been studied theoretically [157] and simulation tools with a fully compressible approach for the gas phase have been reported [94]. Comparison with the numerical results obtained with VULC1D could provide useful insights on the behaviour of the 1D model and, in particular, explore its dynamics in configurations which do not comply with the assumptions used in the aforementioned theoretical study.

## Numerical analysis

The numerical analysis of the unsteady 1D model has permitted the choice of well-suited integration methods. The selected ESDIRK schemes use stiffly accurate embedded methods to improve the error estimates for DAEs. These estimates are however not L-stable, which can lead to unnecessary time step restrictions for stiff problems. Other methods **using main and embedded methods which are both L-stable** could be contemplated, as suggested in [231]. Alternatively, a non-L-stable error estimate can be improved substantially by employing a trick due to Shampine and detailed in [230], where a linear system involving the original error estimate and the Newton iteration matrix is solved to construct an “L-stabilized” version of the estimate.

The typical dimensions (roughly 1000 to 3000 unknowns) of the unsteady 1D model after semi-discretisation in space is not suited for parallelism in space, where the gas and solid phases are split in subdomains. Alternatively, **parallelisation of the Runge-Kutta stages** could be envisioned, an approach also referred to as parallelism across the method. This topic has already been explored and, in particular, parallel versions of fully implicit Runge-Kutta (FIRK) methods [312–314] have been proposed, involving the solution of DIRK-methods on different computer cores. After an iterative process, the solution of the original fully implicit method is obtained, but with the advantage that only sequential DIRK-stages have been solved on each core. This approach could enable better stability and accuracy properties to be obtained, because the parallel iterative process lets the solution converge to that of the original FIRK methods, which can have a higher-stage order (e.g. Radau5) than possible with traditional (ES)DIRK methods. The parallel approach is also very well suited to desktop machines, since the number of cores required is low (typically 2 to 4). However,

such methods have so far not been used on DAEs, and more investigations may be required, in particular regarding the convergence of the parallel iterations.

The **generalisation of the 1D model to higher dimensions** leads to a more complex mathematical structure, since the multidimensionality of the gas velocity field does not allow the momentum equation to be discarded. Thus, a hydrodynamic pressure perturbation needs to be introduced, which leads to an index-2 DAE system overall. This increase of the index usually greatly complicates the solution process [218]. It should therefore be investigated, whether a monolithic approach still is advantageous or not. In particular, it is known that the convergence of a temporal integrator for index-2 DAEs is dictated by the stage order of the method, which is at most 2 for ESDIRK methods. QESDIRK methods [315], which mimic stage order 3, or fully implicit collocation-based methods like Radau5 [206] could be of interest, as has been demonstrated recently [240].

The **stability properties of the adaptive coupling method** presented in Chapter 8 should be investigated, in particular in the context of conjugate heat transfer, to properly derive the stability limits, depending on the spatial mesh refinement and the coupling time step. The **conservativity of the adaptive coupling** should be investigated. Indeed, traditional co-simulation approaches usually do not conserve energy [316]. Therefore, a careful analysis of the mechanisms behind the lack of energy conservation should be conducted, so that corrections can be proposed.

The **application of co-simulation techniques to the toy model** from Section 8.2 should be investigated further, as this model captures the main phenomena involved in conjugate heat transfer problems, with the addition of reactive surface.

Finally, **other extrapolation methods** could be considered for the co-simulation, for example Hermite polynomials [267], which provide approximation orders identical to those of Lagrange polynomials with only half as many sampling points. They however require the specification of the temporal derivatives of the coupling variables, which can be difficult to obtain.

## Improvements of the 1D model and parametrisation

Evolutions of the modelling can also be envisioned for the 1D model from Chapter 2 used in VULC1D. **More physics can be included in the 1D model** to permit the simulation of complex flame structures, e.g. liquid layer at the surface. These changes may require important modifications of the VULC1D code, in particular if the liquid layer is to be spatially discretised.

Besides, it would be interesting to investigate the effect of **other pyrolysis laws**. In particular, in-depth unsteady degradation of the solid phase can be considered, as it is expected to better reproduce the unsteady behaviour of some propellants [163].

Application of VULC1D to the **study of the propellant combustion response to pressure oscillations** could be of interest for other ongoing studies [95] which are focused on the unsteady dynamics of SRMs. Comparison with experimental response data could allow for an **improved determination of the model parameters**, since it provides additional relations that can be used in the optimisation procedure of Chapter 9. Similarly, better propellant model parametrisation may be obtained by including additional experimental measurements, such as temperature sensitivity. Also, **uncertainty quantification** should be addressed based on the estimated precision of the available experimental data, with the aim of obtaining a model parametrisation that is optimal in a statistical sense.

## Improvement of the coupled modelling

Several modelling evolutions can be envisioned for the coupling of the 1D model and the CFD fluid model from CEDRE. **Transition to large-eddy simulations (LES)** may lead to an improved reproduction of the chamber flow and its unsteady character, in particular turbulence-related effects, since the LES approach resolves a broader range of eddies, only modelling the viscous dissipation of much smaller eddies compared to RANS. These smaller eddies have a more general behaviour, thus the approximation error of modelling their effect via a turbulent viscosity is much smaller than in RANS. To avoid the important near-surface mesh refinement required for a proper resolution of the conjugate heat transfer between the chamber flow and the propellant surface, **use of wall laws** in the RANS or LES framework should be investigated, to enable an accurate capture of the conjugate heat transfer with coarser meshes, so as to greatly reduce the computational burden for the CFD solver. Steady-state wall laws such as those from [257] could be adapted. Alternatively, better physical fidelity may be achieved by adding turbulence-related conservation equations in the 1D model from VULC1D so that the 1D model also acts as an unsteady wall law. An accurate representation of the turbulent features inside the propellant flame zone should also enable the reproduction of erosive burning effects [140]. For these aspects, the detailed coupled approach may be of interest. It can indeed be used to **generate wall laws numerically**, by performing a series of simulations in a small domain with different boundary conditions (in particular with respect to turbulence), so as to obtain a tabulation of the wall heat flux depending on the outer flow characteristics.

It would be interesting to find a way to **recover the propellant flame unsteadiness**, which has been discarded in Chapter 7 to ensure the conservativity and temporal coherence of the coupling. The unsteadiness could permit a better capture of interactions between the chamber acoustics and the propellant flame at high frequencies.

Additional physical effects should also be taken into account. Radiative heating can have a large impact on the ignition of propellant areas which are not directly exposed to the igniter flow, for instance in star-shaped regions of the propellant load. To reproduce such a mechanism, in-depth or surface **radiative heating of the propellant surface** by radiant emission from the chamber gas flow and aluminium particles can be easily added to the coupled modelling, using the radiation solvers REA or ASTRE from CEDRE [71] as additional solvers integrated in a split manner with the CFD solver CHARME.

**Aluminium particles** should also be better represented. They can have large impact on the thermal properties of the propellant, and the effect of aluminium fusion and oxidation near the surface should be investigated. Also, aluminium particles are usually rather large (a few dozen microns) as they leave the surface, and this could impact the near-surface flow field and combustion. Whether such effects can be appropriately captured with simple modifications of the 1D model (pyrolysis law, additional source terms...) should be investigated.

Finally, **effects linked to the heterogeneous propellant and flame structures** can be statistically studied through detailed simulations of the burning of a small area with a geometrically-detailed representation of the propellant structure, as was for example studied in [123, 131]. Such studies could be performed with the code COMPAS from ONERA [20]. The obtained statistics can be used to enrich the 1D model and validate its behaviour.

The modelling of the **nozzle seal rupture** during ignition should also be improved. In typical ignition simulations with CEDRE, the nozzle seal is a fictitious wall that is instantaneously removed upon reaching the rupture pressure. Dynamic effects, e.g. pressure fluctuations induced by the seal partially blocking the nozzle flow during its ejection, are not captured. Detailed ignition simulations

should include a model of the seal, its rupture (potentially creating smaller parts), and its interaction with the nozzle flow during its ejection.

The **reproduction of the igniter flow** can also be improved. In particular, the igniter chamber should be appropriately modelled to properly capture the igniter flow start-up and the internal heat losses which lower the energy content transmitted to the main propellant. The assumption of chemical equilibrium for the igniter jet should also be relaxed. Indeed, reactions may influence the shape and dynamics of the jet at the exit of the igniter.

## High-performance computing

To improve the accuracy and computational efficiency of the coupling from Chapter 7, the approach from Chapter 8 for **higher-order adaptive coupling can be implemented in CEDRE** with a reasonable amount of code developments, by introducing a polynomial extrapolation of the coupling variables and controlling the coupling the time step based on the error estimates provided by this extrapolation.

The coupled approach from this manuscript is explicit, and this does not seem to cause stability issues, except when very fine meshes are used. That specific problem disappears when the mesh is less refined, e.g. when wall laws are used in the CFD solver. The stability is indeed largely improved by the fact that the coupling time step is equal to the CFD time step in our current implementation. Computational efficiency could be improved by **enabling subcycling of both VULC1D and CEDRE**. However, instabilities could then reappear if the coupling time step is taken much larger than the CFD time step. Therefore, it would be interesting to work towards **making the coupling implicit**, so as to alleviate these stability constraints. Much work has already been proposed in the co-simulation community regarding that aspect [267, 317–319]. **Working towards more strongly coupled solutions**, e.g. using waveform-relaxation methods [255], would result in a highly-stable solution method. However, if the cost of iterating on a time step with a waveform relaxation approach proves to be too large, non-iterative linearly implicit approaches could be used, but the calculation of the required interface Jacobians (gradients of the state at  $t + \Delta t$  with respect to the inputs for each subsystem) can be cumbersome with complex solvers such as CEDRE. Hence efficient ways of computing them or alternative approaches should be investigated.

## Applications

The coupled approach from Chapter 7 should be applied to **realistic 3D SRM configurations**, so as to assess the adequacy of the chosen modelling level and of the improvements suggested in the previous paragraphs. Well-documented and instrumented test motors are available at ONERA and ArianeGroup, which should be used as test cases for the simulation method.

The propagation speed of the propellant flame along a propellant load should be validated with well-documented test benches, which study this phenomenon in an environment close to that of actual SRMs, via the use of optical diagnostics.

Finally, **applying the adaptive coupling approach from Chapter 8 to other fields** should be considered. In particular, simulations with high order in time for the conjugate heat transfer between a solid and gas flow have not been reported in the literature in the case of a split integration, but only for fully coupled approaches where a single implicit time scheme is used in both solvers, each stage of the scheme being iteratively solved using a waveform relaxation procedure [264, 265], or in a non-iterative manner by handling the coupling terms explicitly (IMEX integration) [266].

# Bibliography

- [1] G.P. Sutton and O. Biblarz. *Rocket propulsion elements*. John Wiley & Sons, 2016.
- [2] Y.B. Zeldovich. On the theory of propellant combustion. *Zhurnal Eksperimental'noi i Teoreticheskoi Fiziki*, 12(11-12):498, 1942.
- [3] F. Culick. Combustion instabilities in solid propellant rocket motors. *Internal aerodynamics in solid rocket propulsion*, 2004. , RTO-EN-023 RTO/NATO.
- [4] V.N. Viliunov and V.E. Zarko. *Ignition of solids*. Studies in physical and theoretical chemistry. Elsevier, 1989.
- [5] R. Hartfield, R. Jenkins, J. Burkhalter, and W. Foster. A review of analytical methods for solid rocket motor grain analysis. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2003.
- [6] A. Davenas. *Technologie des propergols solides*. Masson, 1989.
- [7] W. Dick, R. Fiedler, and H. Heath. Building rocstar: Simulation science for solid propellant rocket motors. In *42nd AIAA/ASME/SAE/ASSE Joint Propulsion Conference and Exhibit*, 2006.
- [8] Q. Li, G.G. He, P.J. Liu, and J. Li. Coupled simulation of fluid flow and propellant burning surface regression in a solid rocket motor. *Computers and Fluids*, 93:146–152, 2014.
- [9] N. Lupoglazoff and F. Vuillot. Numerical simulation of vortex shedding phenomenon in 2d test case solid rocket motors. In *30th Aerospace Sciences Meeting and Exhibit*, 1992.
- [10] F. Chedeveigne. *Instabilités intrinsèques des moteurs à propergol solide*. PhD thesis, École nationale supérieure de l'aéronautique et de l'espace, 2007.
- [11] S. Gallier and F. Godfroy. Aluminum combustion driven instabilities in solid rocket motors. *Journal of propulsion and power*, 25(2):509–521, 2009.
- [12] J. Dupays. *Contribution à l'étude du rôle de la phase condensée dans la stabilité d'un propulseur à propergol solide pour lanceur spatial*. PhD thesis, INP Toulouse, 1996.
- [13] J. Dupays, F. Godfroy, O. Orlandi, P. Prevot, M. Prévost, S. Gallier, S. Ballereau, and Y. Fabignon. Inert condensed phase driving effect of combustion instabilities in solid rocket motors. In *5th International Spacecraft Propulsion Conference, Heraklion, Gr*, page 5, 2008.
- [14] A. Sibra. *Modélisation et étude de l'évaporation et de la combustion de gouttes dans les moteurs à propergol solide par une approche eulérienne Multi-Fluide*. PhD thesis, Université Paris Saclay, 2015.
- [15] V. Dupif. *Modélisation et simulation de l'écoulement diphasique dans les moteurs-fusées à propergol solide par des approches eulériennes polydispersées en taille et en vitesse*. PhD thesis, Université Paris Saclay, 2019.

- [16] A. Guy, B. Fromentin-Denozière, H.K. Phan, A. Cheraly, D. Gueyffier, V. Rialland, C. Erades, P.Q. Elias, J. Labaune, J. Jarrige, A. Ristori, C. Brossard, and S. Rommeluere. Ionized solid propellant rocket exhaust plume: Miles simulation and comparison to experiment. In *7th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2017.
- [17] M. Gross. *Two-Dimensional Modeling of AP/HTPB Utilizing a Vorticity Formulation and One-Dimensional Modeling of AP and ADN*. PhD thesis, Brigham Young University - Provo, 2007.
- [18] M.W. Beckstead, K.V. Puduppakkam, P.R. Thakre, and V. Yang. Modeling of combustion and ignition of solid-propellant ingredients. *Progress in Energy and Combustion Science*, 33(6):497 – 551, 2007.
- [19] T.L. Jackson and J. Buckmaster. Heterogeneous propellant combustion. *AIAA Journal*, 40(6):1122–1130, 2002.
- [20] D. Davidenko and Y. Fabignon. Some aspects of detailed modeling of solid rocket composite propellants. In *6th European Conference for Aeronautics and Space Sciences (EUCASS)*, 06 2015.
- [21] S. Gallier and M. Plaud. Understanding the burning of heterogeneous solid propellants through mesoscale modeling. *International Journal of Energetic Materials and Chemical Propulsion*, 20(2):67–86, 2021.
- [22] T. Sojourner, D.E. Richardson, B.D. Allen, S. Hyde, S. McHenry, B. Goldberg, D. Devries, and M. Ewing. Solid rocket motor reliability and historical failure modes review. In *51st AIAA/SAE/ASEE Joint Propulsion Conference*, 2015.
- [23] Vega VV-15 Flight Investigation Report. <https://www.arianespace.com/press-release/vega-flight-vv15-findings-of-the-independent-inquiry-commissions-investigations/>. Accessed: 2021-15-11.
- [24] E.K. Bastress and W.R. Niessen. Solid propellant ignition by convective heating. Technical report, Little (Arthur D.) Inc Cambridge Mass, 1966.
- [25] S. Boulal, R.W. Devillers, J.Y. Lestrade, J.M. Lamet, C. Corato, D. Henry, and J. Dupays. Experimental investigation on the heat fluxes generated by AP/HTPB solid propellant flames. *Combustion and Flame*, 228:89–98, 2021.
- [26] E.W. Price, H.H. Bradley, G.L. Dehority, and M.M. Ibiricu. Theory of ignition of solid propellants. *AIAA J.*, 4(7):1153–1181, 1966.
- [27] A. Bizot, J. Ferreira, and G. Lengellé. Modelization of the ignition process of homogeneous propellants. In *21st Joint Propulsion Conference*, 07 1985.
- [28] H. Krier, J.S. T'ien, W.A. Sirignano, and M. Summerfield. Nonsteady burning phenomena of solid propellants - theory and experiments. *AIAA journal*, 6(2):278–285, 1968.
- [29] L. De Luca. A critical review of solid rocket propellant transient flame models. *Pure and applied chemistry*, 62(5):825–838, 1990.
- [30] B.V. Novozhilov. *Theory of Nonsteady Burning and Combustion Stability of Solid Propellants by the Zeldovich-Novozhilov Method*, chapter 15, pages 601–641. American Institute of Aeronautics and Astronautics, 1992.
- [31] J.W. Weber, M.Q. Brewster, and K.C. Tang. Radiative ignition and extinction dynamics of energetic solids. *Journal of thermophysics and heat transfer*, 19(3):257–265, 2005.

- [32] Y.C. Liau, E.S. Kim, and V. Yang. A comprehensive analysis of laser-induced ignition of RDX monopropellant. *Combustion and Flame*, 126(3):1680 – 1698, 2001.
- [33] D.A. Smith. *Modeling Solid Propellant Ignition Events*. PhD thesis, Brigham Young University, 2011.
- [34] K. Meredith, M. Gross, and M. Beckstead. Laser-induced ignition modeling of HMX. *Comb. and Flame*, 162, 09 2014.
- [35] M. Jeppson, M. Beckstead, and Q. Jing. A kinetic model for the premixed combustion of a fine AP/HTPB composite propellant. In *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998.
- [36] W. Erikson and M. Beckstead. Modeling unsteady monopropellant combustion with full chemical kinetics. In *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998.
- [37] M.W. Beckstead, V. Yang, and K. Puduppakkam. Modeling and simulation of combustion of solid propellant ingredients using detailed chemical kinetics. In *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2004.
- [38] A.K. Kulkarni, M. Kumar, and K.K. Kuo. Review of solid-propellant ignition studies. *AIAA Journal*, 20(2):243–244, 1982.
- [39] S. Gallier, A. Ferrand, and M. Plaud. Three-dimensional simulations of ignition of composite solid propellants. *Combustion and Flame*, 173:2–15, 11 2016.
- [40] H. Im, L.L. Raja, R.J. Kee, and L.R. Petzold. A numerical study of transient ignition in a counterflow nonpremixed methane-air flame using adaptive time integration. *Combustion Science and Technology*, 158(1):341–363, 2000.
- [41] T. Sayadi, V. Le Chenadec, P.J. Schmid, F. Richecoeur, and M. Massot. Thermoacoustic instability – a dynamical system and time domain analysis. *Journal of Fluid Mechanics*, 753:448–471, 2014.
- [42] K. Parker and M. Summerfield. The ignition transient in solid propellant rocket motors. In *Solid Propellant Rocket Conference*, 1964.
- [43] S. Desoto and H.A. Friedman. Flame spreading and ignition transients in solid grain propellants. *AIAA Journal*, 3(3):405–412, 1965.
- [44] D.M. Adams. Igniter performance in solid-propellant rocket motors. *Journal of Spacecraft and Rockets*, 4(8):1024–1029, 1967.
- [45] R.H. Sforzini and H.L. Fellows. Prediction of ignition transients in solid-propellant rocket motors. *Journal of Spacecraft and Rockets*, 7(5):626–628, 1970.
- [46] A. Peretz, K.K. Kuo, L.H. Caveny, and M. Summerfield. Starting transient of solid-propellant rocket motors with high internal gas velocities. *AIAA Journal*, 11(12):1719–1727, 1973.
- [47] L.H. Caveny, K.K. Kuo, and B.W. Shackelford. Thrust and ignition transients of the Space Shuttle solid rocket motor. *Journal of Spacecraft and Rockets*, 17(6):489–494, 1980.
- [48] L. d’Agostino, L. Biagioni, and G. Lamberti. An ignition transient model for solid propellant rocket motors. In *37th Joint Propulsion Conference and Exhibit*, 2001.
- [49] M. Giacinto, E. Cavallini, B. Favini, and J. Steelant. Parametric study of SRMs design options affecting motor start-up and onset of pressure oscillations. In *5th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2013.

- [50] W.A. Johnston. Solid rocket motor internal flow during ignition. *Journal of Propulsion and Power*, 11(3):489–496, 1995.
- [51] A. Bizot. Ignition and unsteady combustion of AP-based composite propellants in subscale solid rocket motors. *International Journal of Energetic Materials and Chemical Propulsion*, 4(1-6):1046–1061, 1997.
- [52] P. Le Helley. 3D turbulent Navier-Stokes simulations of ignition transients in solid rocket motors. In *34th AIAA/ASME/SAE/ASSE Joint Propulsion Conference and Exhibit*, 1998.
- [53] P. Alavilli, J. Buckmaster, T.L. Jackson, and M. Short. Ignition-transient modeling for solid propellant rocket motors. In *36th AIAA/ASME/SAE/ASSE Joint Propulsion Conference and Exhibit*, 2000.
- [54] R. Fiedler, A. Haselbacher, M. Breitenfeld, P. Alexander, L. Massa, and W. Ross. 3-D Simulations of Ignition Transients in the RSRM. In *41st AIAA/ASME/SAE/ASSE Joint Propulsion Conference and Exhibit*, 2005.
- [55] M. Brandyberry, R. Fiedler, and C. McLay. Verification and validation of the rocstar 3-d multi-physics solid rocket motor simulation program. In *41st AIAA/ASME/SAE/ASSE Joint Propulsion Conference and Exhibit*, 2005.
- [56] Q. Li, P. Liu, and G. He. Fluid–solid coupled simulation of the ignition transient of solid rocket motor. *Acta Astronautica*, 110:180 – 190, 2015. Dynamics and Control of Space Systems.
- [57] Y.K. Li, J.L. Han, X. Chen, R. Liu, and J.S. Xu. Numerical study of the internal flow field of a dual pulse solid rocket motor including conjugate heat transfer. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 231(8):1535–1549, 2017.
- [58] Y.K. Li, X. Chen, J. Xu, C. Zhou, and O. Musa. Three-dimensional multi-physics coupled simulation of ignition transient in a dual pulse solid rocket motor. *Acta Astronautica*, 146:46–65, 2018.
- [59] Y. Ma, M. Zhan, F. Bao, L. Sun, and R. Wei. Numerical simulation of ignition transient for solid rocket motors with large aspect ratio. In *12th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 66–71. IEEE, 2021.
- [60] M. Salita. Modern SRM ignition transient modeling. I-Introduction and physical models. In *37th Joint Propulsion Conference and Exhibit*, 2001.
- [61] A. Bizot. Ignition and unsteady combustion of AP-based composite propellants in subscale solid rocket motors. In *4th Symposium on Special Topics in Chemical Propulsion*, 1996.
- [62] M.J. Ward, S.F. Son, and M.Q. Brewster. Steady deflagration of HMX with simple kinetics: A gas phase chain reaction model. *Combustion and Flame*, 114(3):556–568, 1998.
- [63] M.Q. Brewster. Combustion mechanisms and simplified-kinetics modeling of homogeneous energetic solids. *Energetic Materials: Part 2. Detonation, Combustion*, 13(8):225–294, 2003.
- [64] Ya.B. Zeldovich, G.I. Barenblatt, V.B. Librovich, and G.M. Makhviladze. *The Mathematical Theory of Combustion and Explosion*. Plenum Publishing, 1985.
- [65] M.R. Denison and E. Baum. A simplified model of unstable burning in solid propellants. *ARS Journal*, 31(8):1112–1122, 1961.
- [66] L. François, J. Dupays, D. Davidenko, and M. Massot. Travelling wave mathematical analysis and efficient numerical resolution for a one-dimensional model of solid propellant combustion. *Combustion Theory and Modelling*, 24(5):775–809, 2020.

- [67] L. François, J. Dupays, D. Davidenko, and M. Massot. Travelling wave mathematical analysis and efficient numerical resolution for a realistic model of solid propellant combustion. In *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2019.
- [68] A. Kværnø. Singly diagonally implicit Runge–Kutta methods with an explicit first stage. *BIT*, 44:489–502, 08 2004.
- [69] L. François, J. Dupays, D. Davidenko, and M. Massot. Solid propellant combustion in the low Mach one-dimensional approximation: from an index-one differential-algebraic formulation to high-fidelity simulations through high-order time integration with adaptive time-stepping. hal-02888035 v1, 2021.
- [70] L. François, J. Dupays, and M. Massot. A new simulation strategy for solid rocket motor ignition: coupling a cfd code with a one-dimensional boundary flame model, verification against a fully resolved approach. In *AIAA Propulsion and Energy 2021 Forum*, 2021.
- [71] A. Refloch, B. Courbet, A. Murrone, P. Villedieu, C. Laurent, P. Gilbank, J. Troyes, L. Tessé, G. Chaineray, J.-B. Dargaud, E. Quémerais, and F. Vuillot. Cedre software. *AerospaceLab Journal*, pages 1–10, 2011.
- [72] W.R. Hu. *Space science in China*. CRC Press, 1997.
- [73] J. Guilmartin. The earliest shipboard gunpowder ordnance: An analysis of its technical parameters and tactical capabilities. *The Journal of Military History*, 71:649–669, 07 2007.
- [74] L.T. De Luca. Highlights of solid rocket propulsion history. In *Chemical Rocket Propulsion*, pages 1015–1032. Springer, 2017.
- [75] E. Sir Bullard. The Effect of World War II on the Development of Knowledge in the Physical Sciences [and Discussion]. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 342(1631):519–536, 1975.
- [76] A. Davenas. Development of modern solid propellants. *Journal of Propulsion and Power*, 19(6):1108–1128, 2003.
- [77] A. Davenas. History of the development of solid rocket propellant in France. *Journal of Propulsion and Power*, 11(2):285–291, 1995.
- [78] S. Burroughs. Status of army pintle technology for controllable thrust propulsion. In *37th Joint Propulsion Conference and Exhibit*, 2001.
- [79] S.L. Burroughs, R.S. Michaels, W.L. Alford, A.B. Spencer, and K.L. Peterson. Demonstration of a pintle controlled solid rocket motor in a tactical mission application. In *1996 JANNAF Propulsion Meeting, Albuquerque, New Mexico*, 1996.
- [80] M. Ostrander, J. Bergmans, M. Thomas, and S. Burroughs. Pintle motor challenges for tactical missiles. In *36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3310, 2000.
- [81] J. Sapkota, Y.H. Xu, and H.J. Sun. Numerical study on response characteristics of solid rocket pintle motor. *Journal of Aerospace Technology and Management*, 11, 2019.
- [82] H. Dahl and B. Jones. Demonstration of solid propellant pulse motor technologies. In *32nd Joint Propulsion Conference and Exhibit*, 1996.
- [83] B.J. Cantwell. AA283: Aircraft and Rocket Propulsion. Stanford University, 2021.
- [84] K. Tsiolkovsky. Investigation of outer space rocket appliances (Issledovanie mirovyh prostanstv reaktivnymi priborami). *Naučnoe obozrenie*, 5, 1903.

- [85] A.I. Atwood, T.L. Boggs, P.O. Curran, T.P. Parr, D.M. Hanson-Parr, C.F. Price, and J. Wiknich. Burning rate of solid propellant ingredients, part 1: Pressure and initial temperature effects. *Journal of Propulsion and Power*, 15(6):740–747, 1999.
- [86] L. François. personal GitHub page, <https://github.com/laurent90git/>.
- [87] J.I. Schafer. *Solid Rocket Propulsion*. H.F. Seifert, John Wiley and Sons, New-York, space technology edition, 1959.
- [88] M. Calabro, P. Perot, G. Joannis, and G. Vigier. Reverse forward dome for a missile first stage. In *23rd Joint Propulsion Conference*, page 1989, 1987.
- [89] M. Prevost, P. Jezequel, and C. Corato. Étude des oscillations de poussée - t11.3 synthèse de la base de données expérimentales instationnaires, version 8. rapport technique. Technical Report RT 20/19330, ONERA, 2013.
- [90] F.A. Williams, M. Barrere, and N.C. Huang. *Fundamental aspects of solid propellant rockets*. Advisory Group for Aerospace Research and Development of NATO, Technivision Services; London, 1969.
- [91] P. Kuentzmann. Modèles de combustion instationnaire des propergols solides composites. *La Recherche Aéronautique*, 2:77–93, 1978.
- [92] F. Vuillot and N. Lupoglazoff. Combustion and turbulent flow effects in 2d unsteady navier-stokes simulations of oscillatory solid rocket motors - first applications. In *34th Aerospace Sciences Meeting and Exhibit*, 1996.
- [93] Y. Fabignon, J. Dupays, G. Avalon, F. Vuillot, N. Lupoglazoff, G. Casalis, and M. Prévost. Instabilities and pressure oscillations in solid rocket motors. *Aerospace Science and Technology*, 7(3):191–200, 2003.
- [94] S. Rahman. *Modélisation et simulation numérique de flammes planes instationnaires de perchlorate d’ammonium*. PhD thesis, Université Pierre et Marie Curie, 2012.
- [95] M. Carricart. *Modélisation de la combustion des matériaux énergétiques nouvelle génération*. PhD thesis, ISAE Supaéro, 2023. in preparation.
- [96] R.S. Brown, R. Dunlap, S.W. Young, and R.C. Waugh. Vortex shedding as a source of acoustic energy in segmented solid rockets. *Journal of Spacecraft and Rockets*, 18(4):312–319, 1981.
- [97] H. Nomoto and F.E. Culick. An experimental investigation of pure tone generation by vortex shedding in a duct. *Journal of Sound and Vibration*, 84(2):247–252, 1995.
- [98] A. Flatau and W. Vanmoorhem. Prediction of vortex shedding responses in segmented solid rocket motors. In *26th Joint Propulsion Conference*, 1990.
- [99] F. Vuillot. Vortex-shedding phenomena in solid rocket motors. *Journal of Propulsion and Power*, 11(4):626–639, 1995.
- [100] G.A. Flandro. Vortex driving mechanism in oscillatory rocket flows. *Journal of Propulsion and Power*, 2(3):206–214, 1986.
- [101] J. Traineau, M. Prevost, F. Vuillot, P. Le Breton, J. Cuny, N. Preioni, R. Bec, J. Traineau, M. Prevost, F. Vuillot, P. Le Breton, J. Cuny, N. Preioni, and R. Bec. A subscale test program to assess the vortex shedding driven instabilities in segmented solid rocket motors. In *33rd Joint Propulsion Conference and Exhibit*, 1997.
- [102] G. Casalis. Stabilité linéaire d’un écoulement dans un conduit à parois débitantes en géométrie cartésienne. Technical Report RF 114/5118.25, ONERA, 1998.

- [103] F. Vuillot. Recent advances on the stability of large segmented space boosters. In *2nd European Conference on Launcher Technology, Space Solid Propulsion*, 2000.
- [104] S. Gallier, F. Godfroy, and F. Plourde. Computational study of turbulence in a subscale solid rocket motor. In *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 07 2004.
- [105] K.W. Dotson and B.H. Sako. Interaction between solid rocket motor internal flow and structure during flight. *Journal of Propulsion and Power*, 23(1):140–145, 2007.
- [106] I.S. Chang, N. Patel, and S. Yang. Titan IV SRMU anomaly and redesign analyses. In *30th Joint Propulsion Conference and Exhibit*, 1994.
- [107] A Namazifard, K. Hjelmstad, P. Sofronis, K. Nakshatrala, D. Tortorelli, and R. Fiedler. Simulations of propellant slumping in the Titan IV SRMU using constitutive models with damage evolution. In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2005.
- [108] J. Richard. *Développement d’une chaîne de calcul pour les interactions fluide-structure et applications aux instabilités aéroacoustiques d’un moteur à propergol solide*. PhD thesis, Université de Montpellier II, 2013.
- [109] A.F. Belyaev, Y.V Frolov, and A.I. Korotkov. Combustion and ignition of particles of finely dispersed aluminum. *Combustion, Explosions and Shock Waves*, 4:182–185, 1968.
- [110] Y. Fabignon, J.F. Trubert, D. Lambert, O. Orlandi, and J. Dupays. Combustion of aluminum particles in solid rocket motors. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2003.
- [111] M.W. Beckstead. Correlating aluminum burning times. *Combustion, Explosion, Shock Waves*, 41:533–546, 09 2005.
- [112] M. Muller. *Modélisation de la combustion de gouttes d’aluminium dans les conditions d’un moteur fusée à propergol solide*. PhD thesis, Sorbonne Université, 2019.
- [113] A. Braconnier, G. Stany, H. Fabien, and Christian C. Aluminum combustion in CO<sub>2</sub>-CO-N<sub>2</sub> mixtures. *Proceedings of the Combustion Institute*, 38(3):4355–4363, 2021.
- [114] P.H. Chevalier, N. Dorval, R.W. Devillers, D. Davidenko, X. Mercier, and J. Pichillou. Investigation of aluminum droplet combustion in solid propellant flames: Al-PLIF experiments and numerical simulation. In *AIAA Propulsion and Energy 2021 Forum*.
- [115] F. Doisneau. *Eulerian modeling and simulation of polydisperse moderately dense coalescing spray flows with nanometric-to-inertial droplets : application to Solid Rocket Motors*. PhD thesis, École Centrale Paris, 2013.
- [116] A. Genot, S. Gallier, and T. Schuller. Model for acoustic induced aluminum combustion fluctuations in solid rocket motors. *Journal of Propulsion and Power*, 35(4):720–735, 2019.
- [117] G. Genot. Aluminum combustion instabilities: Dimensionless numbers controlling the instability in solid rocket motors. *Combustion and Flame*, 232:111563, 2021.
- [118] M.P. Gale, M.F. Alam, and E.A. Luke. A multiphase modeling framework for solid rocket motor slag accumulation and dynamics prediction. In *AIAA Propulsion and Energy 2021 Forum*.
- [119] V. Giovangigli, N. Meynet, and M.D. Smooke. Application of continuation techniques to ammonium perchlorate plane flames. *Combustion Theory and Modelling*, 10(5):771–798, 2006.

- [120] L. Massa, T.L. Jackson, and J. Buckmaster. New kinetics for a model of heterogeneous propellant combustion. *Journal of Propulsion and Power*, 21(5):914–924, 2005.
- [121] B.T. Bojko, M. Gross, and T.L. Jackson. Investigating dimensional effects on predicting burning rates of heterogeneous solid propellants. In *AIAA Scitech 2019 Forum*.
- [122] S. Gallier. Heterogeneous solid propellants: from microstructure to macroscale properties. *Progress in Propulsion Physics*, 2:21–34, 2011.
- [123] S. Gallier and M. Plaud. A model for solid propellant burning fluctuations using mesoscale simulations. *Acta Astronautica*, 158:296–303, 2019.
- [124] D. Davidenko and Y. Fabignon. Modeling of heterogeneous structure and properties of composite solid propellants. In *5th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2013.
- [125] L. Massa, T.L. Jackson, J. Buckmaster, and F. Najjar. Fluctuations above a burning heterogeneous propellant. *Journal of Fluid Mechanics*, 581:1–32, 2007.
- [126] M. Mehrabadi, M. Safdari, W. Dick, and T.L. Jackson. Effect of pressure on the fluctuations above a burning heterogeneous propellant. In *AIAA Scitech 2021 Forum*.
- [127] Y. Daimon, T.L. Jackson, V. Topalian, J.B. Freund, and J. Buckmaster. Effect of propellant morphology on acoustics in a planar rocket motor. *Theoretical and Computational Fluid Dynamics*, 23(1):63–77, 2009.
- [128] M. Chen, J. Buckmaster, T.L. Jackson, and L. Massa. Homogenization issues and the combustion of heterogeneous solid propellants. *Proceedings of the Combustion Institute*, 29:2923–2929, 05 2002.
- [129] S. Gallier and J.F. Guery. Regression fronts in random sphere packs: Application to composite solid propellant burning rate. *Proceedings of the Combustion Institute*, 32(2):2115–2122, 2009.
- [130] M.W. Beckstead, R.L. Derr, and C.F. Price. A model of composite solid-propellant combustion based on multiple flames. *AIAA Journal*, 8(12):2200–2207, 1970.
- [131] L. Massa, T.L. Jackson, and J. Buckmaster. Using Heterogeneous Propellant Burning Simulations as Subgrid Components of Rocket Simulations. *AIAA journal*, 42(9):1889–1900, 2004.
- [132] M.L. Gross, T.D. Hedman, S.F. Son, T.L. Jackson, and M.W. Beckstead. Coupling micro and meso-scale combustion models of AP/HTPB propellants. *Combustion and flame*, 160(5):982–992, 2013.
- [133] M.L. Gross and M.W. Beckstead. Steady-state combustion mechanisms of ammonium perchlorate composite propellants. *Journal of Propulsion and Power*, 27(5):1064–1078, 2011.
- [134] M. Plaud, S. Gallier, and M. Morel. Simulations of heterogeneous propellant combustion: Effect of particle orientation and shape. *Proceedings of the Combustion Institute*, 35(2):2447–2454, 2015.
- [135] T.L. Jackson. Issues related to heterogeneous solid-propellant combustion. *Progress in Propulsion Physics*, 2:3–20, 2011.
- [136] T.L. Jackson. Modeling of heterogeneous propellant combustion: A survey. *AIAA journal*, 50(5):993–1006, 2012.
- [137] K.C. Brown, V. Sankaran, and T.L. Jackson. Mesoscale modeling of solid propellant burn rates. In *AIAA Scitech 2019 Forum*, page 1236, 2019.

- [138] K. Kinefuchi, K. Okita, I. Funaki, and T. Abe. Prediction of in-flight radio frequency attenuation by a rocket plume. *Journal of Spacecraft and Rockets*, 52(2):340–349, 2015.
- [139] M. Kumar and K.K. Kuo. Flame spreading and overall ignition transient. *Fundamentals of Solid Propellant Combustion*, 90:305–360, 1984.
- [140] G. Lengelle. Model describing the erosive combustion and velocity response of composite propellants. *AIAA journal*, 13(3):315–322, 1975.
- [141] M.K. Razdan and K.K. Kuo. Erosive burning study of composite solid propellants by turbulent boundary-layer approach. *AIAA Journal*, 17(11):1225–1233, 1979.
- [142] J.H. Frazer and B.L. Hicks. The thermal theory of ignition of solid propellants. *The Journal of Physical and Colloid Chemistry*, 54(6):872–876, 1950.
- [143] R. Anderson, R. Brown, and L. Shannon. Ignition theory of solid propellants. In *Solid Propellant Rocket Conference*, 1964.
- [144] C.E. Hermance, R. Shinnar, and M. Summerfield. Ignition of an evaporating fuel in a hot, stagnant gas containing an oxidizer. *AIAA Journal*, 3(9):1584–1592, 1965.
- [145] W. Erikson and M.W. Beckstead. Modeling pressure and heat flux responses of nitramine monopropellants with detailed chemistry. In *35th Joint Propulsion Conference and Exhibit*, 1999.
- [146] Y. Liao and V. Yang. Analysis of RDX monopropellant combustion with two-phase subsurface reactions. *Journal of Propulsion and Power*, 11(4):729–739, 1995.
- [147] J. Davidson and M.W. Beckstead. Improvements to steady-state combustion modeling of cyclotrimethylenetrinitramine. *Journal of Propulsion and Power*, 13:375–383, 1997.
- [148] L. Massa and T.L. Jackson. Multidimensional numerical simulation of ammonium-perchlorate-based propellant combustion with fine/ultrafine aluminum. *Journal of Propulsion and Power*, 24:161–174, 03 2008.
- [149] L. Massa, T.L. Jackson, J. Buckmaster, and M. Campbell. Three-dimensional heterogeneous propellant combustion. *Proceedings of the Combustion Institute*, 29(2):2975–2983, 2002.
- [150] J.M. Lenoir and G. Robillard. A mathematical method to predict the effects of erosive burning in solid-propellant rockets. In *Symposium (International) on Combustion*, volume 6, pages 663–667. Elsevier, 1957.
- [151] G. Luke, M. Eagar, and H. Dwyer. Ignition transient model for large aspect ratio solid rocket motors. In *32nd Joint Propulsion Conference and Exhibit*, 1996.
- [152] J. Godil and A. Kamran. Numerical simulation of ignition transient in solid rocket motor: a revisit. *Aircraft Engineering and Aerospace Technology*, 2017.
- [153] A. Bizot. Ignition and unsteady combustion of AP-based composite propellants in subscale solid rocket motors. *International Journal of Energetic Materials and Chemical Propulsion*, 4(1-6):1046–1061, 1997.
- [154] Illinois Rocstar. <https://illinoisrocstar.com/>. Accessed: 2021-27-10.
- [155] O. Orlandi, F. Fourmeaux, and J. Dupays. Ignition study at small-scale solid rocket motor. In *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2019.
- [156] F. Culick. A review of calculations for unsteady burning of a solid propellant. *AIAA Journal*, 6(12):2241–2255, 1968.

- [157] P. Clavin and D. Lazimi. Theoretical analysis of oscillatory burning of homogeneous solid propellant including non-steady gas phase effects. *Combustion Science and Technology*, 83(1-3):1–32, 1992.
- [158] L. De Luca. *Theory of Nonsteady Burning and Combustion Stability of Solid Propellants by Flame Models*, chapter 14, pages 519–600. American Institute of Aeronautics and Astronautics, 1992.
- [159] G. Lengellé, J. Duterque, and J.F. Trubert. Physico-chemical mechanisms of solid propellant combustion. *Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics*, 185:287–334, 01 2000-01.
- [160] M. Tanaka and M.W. Beckstead. A three-phase combustion model of ammonium perchlorate. *32nd Joint Propulsion Conference and Exhibit*, page 2888, 1996.
- [161] J. Davidson and M.W. Beckstead. A three-phase model of HMX combustion. *Symposium (International) on Combustion*, 26(2):1989–1996, 1996.
- [162] J. Davidson and M.W. Beckstead. Improvements in RDX combustion modeling. *32nd JAN-NAF Combustion Meeting*, 1:41–56, 01 1995.
- [163] T. Jackson, L. Massa, and M.Q. Brewster. Unsteady combustion modelling of energetic solids, revisited. *Combustion Theory and Modelling*, 8:513–532, 09 2004.
- [164] M. Gross, K. Meredith, and M.W. Beckstead. Fast cook-off modeling of HMX. *Comb. and Flame*, 162(9):3307 – 3315, 2015.
- [165] M.W. Tanner. *Multidimensional Modeling of Solid Propellant Burning Rates and Aluminum Agglomeration and One-Dimensional Modeling of RDX/GAP and AP/HTPB*. PhD thesis, Brigham Young University, 2008.
- [166] P. Bernigaud. *Modélisation de la combustion des matériaux énergétiques nouvelle génération*. PhD thesis, Institut Polytechnique de Paris, 2023. in preparation.
- [167] N. Meynet. *Simulation numérique de la combustion d’un propeergol solide*. PhD thesis, Université Pierre et Marie Curie, 2005.
- [168] V. Giovangigli and A. Ern. EGLib: A Multicomponent Transport Software for Fast and Accurate Evaluation Algorithms. *École polytechnique*, 2004. <http://www.cmap.polytechnique.fr/www.eglib/>.
- [169] B. Desjardins and C.K. Lin. A survey of the Compressible Navier-Stokes equations. *Tawainese Journal of Mathematics*, 3(2):123–137, 1999.
- [170] J.O. Hirschfelder, C.F. Curtiss, and R.B. Bird. *Molecular theory of gases and liquids*. Wiley, New York, 1954.
- [171] V. Giovangigli. Convergent iterative methods for multicomponent diffusion. *IMPACT of Computing in Science and Engineering*, 3(3):244–276, 1991.
- [172] J. Wang, J.K. Carson, M.F. North, and D.J. Cleland. A new structural model of effective thermal conductivity for heterogeneous materials with co-continuous phases. *International Journal of Heat and Mass Transfer*, 51(9):2389–2397, 2008.
- [173] A. Eucken. Allgemeine Gesetzmäßigkeiten für das Wärmeleitvermögen verschiedener Stoffarten und Aggregatzustände. *Forschung auf dem Gebiet des Ingenieurwesens A*, 11:6–20, 1940.

- [174] R. Landauer. Electrical conductivity in inhomogeneous media. *AIP Conference Proceedings*, 40(1):2–45, 1978.
- [175] A.G. Merzhanov and F.I. Dubovitsky. *On the theory of steady-state monopropellant combustion*, volume 129, pages 153–156. USSR Academy of Sciences, 1959.
- [176] B.I. Khaikin and A.G. Merzhanov. Theory of thermal propagation of a chemical reaction front. *Combustion, Explosion and Shock Waves*, 2(3):22–27, 1966.
- [177] A.G. Merzhanov. The theory of stable homogeneous combustion of condensed substances. *Combustion and Flame*, 13(2):143–156, 1969.
- [178] G. Lengellé. Thermal degradation kinetics and surface pyrolysis of vinyl polymers. *AIAA Journal*, 8(11):1989–1996, 1970.
- [179] A.G. Merzhanov. Solid flames: Discoveries, concepts, and horizons of cognition. *Combustion Science and Technology*, 98(4-6):307–336, 1994.
- [180] A.G. Merzhanov and B.I. Khaikin. Theory of combustion waves in homogeneous media. *Progress in energy and combustion science*, 14(1):1–98, 1988.
- [181] P. Clavin and G. Searby. *Combustion waves and fronts in flows: flames, shocks, detonations, ablation fronts and explosion of stars*. Cambridge University Press, 2016.
- [182] M.A. Zebrowski and M.Q. Brewster. Theory of unsteady combustion of solids - investigation of quasisteady assumption. *Journal of Propulsion and Power*, 12(3):564–573, 1996.
- [183] F.A. Williams M.M. Ibricu. Influence of externally applied thermal radiation on the burning rates of homogeneous solid propellants. *Combustion and Flame*, 24:185–198, 1975.
- [184] A. Zenin and B. Novozhilov. Single-valued dependence of the surface temperature of ballistite on the burning rate. *Combustion, Explosion and Shock Waves*, 9, 03 1973.
- [185] A.P. Glazkova, A.A. Zenin, and A.A. Balepin. Influence of initial temperature on parameters of combustion zone of ammonium perchlorate. *Archiwum Procesow Spalania*, 1(3-4):261–268, 1970.
- [186] M.Q. Brewster and S.F. Son. Quasi-steady combustion modeling of homogeneous solid propellants. *Combustion and Flame*, 103(1-2):11–26, 1995.
- [187] L. De Luca, M. Verri, F. Cozzi, G. Colombo, and E. Volpe. Adiabatic pyrolysis of solid propellants. In *35th Aerospace Sciences Meeting and Exhibit*, 1997.
- [188] K.R. Anil Kumar and K.N. Lakshmisha. Dynamic combustion of solid propellants: Effects of unsteady condensed phase degradation layer. *Journal of Propulsion and Power*, 18(2):312–321, 2002.
- [189] V.B. Novozhilov and B.V. Novozhilov. *Theory of Solid-Propellant Nonsteady Combustion*. Wiley, 2020.
- [190] F.A. Williams. Quasi-steady gas-phase flame theory in unsteady burning of a homogeneous solid propellant. *AIAA Journal*, 11(9):1328–1330, 1973.
- [191] M.W. Beckstead, R.L. Derr, and C.F. Price. The combustion of solid monopropellants and composite propellants. *Symposium (International) on Combustion*, 13(1):1047–1056, 1971. Thirteenth symposium (International) on Combustion.
- [192] H. Krier, J.S. T'ien, W.A. Sirignano, and M. Summerfield. Nonsteady burning phenomena of solid propellants - theory and experiments. *AIAA Journal*, 6(2):278–285, 1968.

- [193] M. Schatzman. *Numerical Analysis: A Mathematical Introduction*, volume 140. Oxford University Press, 2002.
- [194] J. Henderson and H. Wang. Positive solutions for nonlinear eigenvalue problems. *J. Math. Anal. Appl.*, 208(1):252–259, 1997.
- [195] M. Verri. Asymptotic stability of traveling waves in solid-propellant combustion under thermal radiation. *Mathematical Models and Methods in Applied Sciences*, 09:1279–1305, 1999.
- [196] W.E. Johnson and W. Nachbar. Laminar flame theory and the steady, linear burning of a monopropellant. *Archive for Rational Mechanics and Analysis*, 12:58–92, 1963.
- [197] H. Berestycki, B. Nicolaenko, and B. Scheurer. Traveling wave solutions to combustion models and their singular limits. *SIAM Journal on Mathematical Analysis*, 16(6):1207–1242, 1985.
- [198] V. Giovangigli. Plane laminar flames with multicomponent transport and complex chemistry. *Mathematical Models and Methods in Applied Sciences*, 9(3):337–378, 1999.
- [199] M.S. Miller. In search of an idealized model of homogeneous solid propellant combustion. *Combustion and Flame*, 46:51–73, 1982.
- [200] W.E. Johnson and W. Nachbar. Deflagration limits in the steady linear burning of a monopropellant with application to ammonium perchlorate. *Symposium (International) on Combustion*, 8(1):678–689, 1961. Eighth Symposium (International) on Combustion.
- [201] T. von Kármán and G. Barbany. The thermal theory of constant pressure deflagration. In *Anniversary Volume on Applied Mechanics dedicated to C. B. Biezeno by some of his Friends and Former Students on the Occasion of his Sixty-Five Birthday, March 2, 1953*, pages 58–69. H. Stam, 1953.
- [202] F.E. Marble. Flame theory and combustion technology. *Journal of the Aeronautical Sciences*, 23(5):462–468, 1956.
- [203] H. Berestycki, B. Larrouturou, and J.M. Roquejoffre. Mathematical investigation of the cold boundary difficulty in flame propagation theory. In P.C. Fife, Amable Liñán, and F. Williams, editors, *Dynamical Issues in Combustion Theory*, pages 37–61. Springer New York, 1991.
- [204] H. Berestycki, L. Nirenberg, and S.R.S. Varadhan. The principal eigenvalue and maximum principle for second-order elliptic operators in general domains. *Communications on Pure and Applied Mathematics*, 47(1):47–92, 1994.
- [205] A.I. Volpert, V.A. Volpert, and V.A. Volpert. *Traveling Wave Solutions of Parabolic Systems: Translations of Mathematical Monographs*, volume 140. American Mathematical Society, 1994.
- [206] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Comput. Math.* Springer-Verlag Berlin Heidelberg, 2nd edition, 1996.
- [207] R.P. Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.
- [208] M.D. Smooke and V. Giovangigli. Numerical modeling of axisymmetric laminar diffusion flames. *IMPACT of Computing in Science and Engineering*, 4(1):46–79, 1992.
- [209] M. Muller, D. Davidenko, and V. Giovangigli. Computational study of aluminum droplet combustion in different atmospheres. In *7th European Conference for Aeronautics and Space Sciences (EUCASS)*, pages 1–17, 07 2017.
- [210] R.J. Kee, F.M. Rupley, and J.A. Miller. Chemkin-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 1989.

- [211] R.J. Kee, J.F. Grcar, M.D. Smooke, J.A. Miller, and E. Meeks. Premix: a Fortran program for modeling steady laminar one-dimensional premixed flames. *Sandia Rep*, 143, 01 1985.
- [212] N. Darabiha. Transient behaviour of laminar counterflow hydrogen-air diffusion flames with complex chemistry. *Comb. Sci. and Tech*, 86:163–181, 1992.
- [213] M. Massot, M. Kumar, A. Gomez, and M. D. Smooke. Counterflow spray diffusion flames of heptane: computations and experiments. In *Proceedings of the 27th Symp. on Comb., The Comb. Institute*, pages 1975–1983, 1998.
- [214] S. Descombes, M. Duarte, T. Dumont, F. Laurent, V. Louvet, and M. Massot. Analysis of operator splitting in the nonasymptotic regime for nonlinear reaction-diffusion equations. Application to the dynamics of premixed flames. *SIAM J. Numer. Anal.*, 52(3):1311–1334, 2014.
- [215] F. Harlow and A. Amsden. A numerical fluid dynamics calculation method for all flow speeds. *Journal of Computational Physics*, 8(2):197 – 213, 1971.
- [216] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: A variable-coefficient ODE solver. *SIAM J. Sci. Stat. Comput.*, 10(5):1038–1051, September 1989.
- [217] J. Grcar. The twopnt program for boundary value problems. Technical report, Sandia National Labs., Livermore, CA, 04 1992.
- [218] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1995.
- [219] S. Balay et al. PETSc Web page. <https://www.mcs.anl.gov/petsc>, 2019.
- [220] M.A. N’Guessan, M. Massot, L. Séries, and C. Tenaud. High order time integration and mesh adaptation with error control for incompressible Navier–Stokes and scalar transport resolution on dual grids. *Journal of Computational and Applied Mathematics*, 387:112542, 2021.
- [221] A. Jameson and E. Turkel. Implicit schemes and LU decompositions. *Mathematics of Computation*, 37(156):385–397, 1981.
- [222] D. Anderson, J.C. Tannehill, and R.H. Pletcher. *Computational fluid mechanics and heat transfer, Third edition*. Boca Raton, CRC Press, 2013.
- [223] P.K. Khosla and S.G. Rubin. Filtering of non-linear instabilities. *Journal of engineering mathematics*, 13(2):127–141, 1979.
- [224] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff problems*. Springer, Berlin, second edition, 2000.
- [225] G.G. Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, 1963.
- [226] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proc. of the Cambridge Philosophical Society*, 43(1):50–67, 1947.
- [227] B.L. Ehle. *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*. PhD thesis, University of Waterloo Waterloo, Ontario, 1969.
- [228] L. Jay. Convergence of a class of runge-kutta methods for differential-algebraic systems of index 2. *BIT Numerical Mathematics*, 33(1):137–150, 1993.

- [229] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff O.D.E.s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.
- [230] M.E. Hosea and L.F. Shampine. Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics*, 20(1-2):21–37, 1996.
- [231] K. Gustafsson. Control-theoretic techniques for stepsize selection in implicit Runge-Kutta methods. *ACM Transactions on Mathematical Software (TOMS)*, 20(4):496–517, 1994.
- [232] G. Söderlind and L. Wang. Adaptive time-stepping and computational stability. *Journal of Computational and Applied Mathematics*, 185(2):225–243, 2006.
- [233] U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.
- [234] S. Descombes, M. Duarte, T. Dumont, V. Louvet, and M. Massot. Adaptive time splitting method for multi-scale evolutionary partial differential equations. *Confluentes Mathematici*, 03(03):413–443, 2011.
- [235] S. Descombes, M. Duarte, and M. Massot. *Operator Splitting Methods with Error Estimator and Adaptive Time-Stepping. Application to the Simulation of Combustion Phenomena*, pages 627–641. Springer International Publishing, 2016.
- [236] E. Motheau and J. Abraham. A high-order numerical algorithm for DNS of low-Mach-number reactive flows with detailed chemistry and quasi-spectral accuracy. *Journal of Computational Physics*, 313:430 – 454, 2016.
- [237] O.M. Knio, H.N. Najm, and P.S. Wyckoff. A semi-implicit numerical scheme for reacting flow. II. Stiff, operator-split formulation. *J. Comput. Phys.*, 154:482–467, 1999.
- [238] H.N. Najm and O.M. Knio. Modeling Low Mach number reacting flow with detailed chemistry and transport. *J. Scientific Computing*, 25(1/2):263–287, 2005.
- [239] P. Virtanen et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [240] M.A. N’Guessan. *Space adaptive methods with error control based on adaptive multiresolution for the simulation of low-Mach reactive flows*. PhD thesis, Université Paris-Saclay, 2020.
- [241] C. Koren, R. Vicquelin, and O. Gicquel. Self-adaptive coupling frequency for unsteady coupled conjugate heat transfer simulations. *International Journal of Thermal Sciences*, 118:340 – 354, 2017.
- [242] E. Radenac, J. Gressier, and P. Millan. Methodology of numerical coupling for transient conjugate heat transfer. *Computers and Fluids*, 100:95 – 107, 2014.
- [243] R. Moretti. *Étude et amélioration des méthodologies de couplage aérothermique fluide-structure*. PhD thesis, Université Paris Sciences et Lettres, 2019.
- [244] T. Poinso and D. Veynante. *Theoretical and numerical combustion*. RT Edwards, Inc., 2005.
- [245] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [246] A.N. Kolmogorov. Equations of turbulent motion of an incompressible fluid. *Izv Akad. Nauk SSR, Seria Fiz. VI*, 1:56–58, 1942.
- [247] W.P. Jones and B.E. Launder. The prediction of laminarization with a two-equation model of turbulence. *International journal of heat and mass transfer*, 15(2):301–314, 1972.

- [248] F. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32:1598–1605, 1994.
- [249] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.
- [250] P. Roe. Approximate Riemann Solvers, Parameter Vector, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 10 1981.
- [251] C. Le Touze. *Couplage entre modèles diphasiques à “phases séparées” et à “phase dispersée” pour la simulation de l’atomisation primaire en combustion cryotechnique*. PhD thesis, Université Nice Sophia Antipolis, 2015.
- [252] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [253] CEDRE. <https://cedre.onera.fr/>. Accessed: 2021-01-11.
- [254] J. White, F. Odeh, A.L. Sangiovanni-Vincentelli, and A. Ruehli. Waveform relaxation: Theory and practice. Technical Report UCB/ERL M85/65, EECS Department, University of California, Berkeley, Jul 1985.
- [255] M.J. Gander. Waveform relaxation. <https://www.unige.ch/~gander/Preprints/EncyclopediaSpringer.pdf>.
- [256] M.B. Giles. Stability analysis of numerical interface conditions in fluid-structure thermal analysis. *International journal for numerical methods in fluids*, 25(4):421–436, 1997.
- [257] A. Bizot. Turbulent boundary layer with mass transfer and pressure gradient in solid propellant rocket motors. In *31st Joint Propulsion Conference and Exhibit*, 1995.
- [258] L. Lees. Convective heat transfer with mass addition and chemical reactions. In *Combustion and Propulsion, Third AGARD Colloquium*, pages 451–498. Pergamon Press, New York, 1958.
- [259] M.M. Hopkins, H.K. Moffat, B. Carnes, R.W. Hooper, and R.P. Pawlowski. Final report on LDRD project : coupling strategies for multi-physics applications. Technical report, Sandia National Laboratories, 2007.
- [260] W. Liniger and R.A. Willoughby. Efficient integration methods for stiff systems of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 7(1):47–66, 1970.
- [261] M.J. Gander, Y.L. Jiang, and R.J. Li. Parareal schwarz waveform relaxation methods. In *Domain decomposition methods in science and engineering XX*, pages 451–458. Springer, 2013.
- [262] M.J. Gander, F. Kwok, and B.C. Mandal. Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems. *Electronic Transactions on Numerical Analysis*, 45:425–456, 2016.
- [263] P. Birken, K.J. Quint, S. Hartmann, and A. Meister. A time-adaptive fluid-structure interaction method for thermal coupling. *Computing and visualization in science*, 13(7):331–340, 2010.
- [264] V. Kazemi-Kamyab, A.H. Van Zuijlen, and H. Bijl. Analysis and application of high order implicit Runge-Kutta schemes for unsteady conjugate heat transfer: A strongly-coupled approach. *Journal of Computational Physics*, 272:471–486, 2014.
- [265] P. Birken, T. Gleim, D. Kuhl, and A. Meister. Fast solvers for unsteady thermal fluid structure interaction. *International Journal for Numerical Methods in Fluids*, 79(1):16–29, 2015.

- [266] V. Kazemi-Kamyab, A.H. Van Zuijlen, and H. Bijl. A high order time-accurate loosely-coupled solution algorithm for unsteady conjugate heat transfer problems. *Computer Methods in Applied Mechanics and Engineering*, 264:205–217, 2013.
- [267] M. Busch. *Zur effizienten Kopplung von Simulationsprogrammen*. PhD thesis, Universität Kassel, 2012.
- [268] M. Duarte. *Méthodes numériques adaptives pour la simulation de la dynamique de fronts de réaction multi-échelle en temps et en espace*. PhD thesis, École Centrale Paris, 2011.
- [269] G. Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.
- [270] H. Yoshida. Construction of higher order symplectic integrators. *Physics letters A*, 150(5-7):262–268, 1990.
- [271] M. Arnold, C. Clauß, and T. Schierz. Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation V2. 0. In *Progress in Differential-Algebraic Equations*, pages 107–125. Springer, 2014.
- [272] T. Meyer, J. Kraft, and B. Schweizer. Co-simulation: Error estimation and macro-step size control. *Journal of Computational and Nonlinear Dynamics*, 16(4):041002, 2021.
- [273] C. Gomes, C. Thule, D. Broman, P.G. Larsen, and H. Vangheluwe. Co-simulation: A survey. *ACM Comput. Surv.*, 51(3), May 2018.
- [274] R. Kübler and W. Schiehlen. Two methods of simulator coupling. *Mathematical and computer modelling of dynamical systems*, 6(2):93–113, 2000.
- [275] M. Busch. Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 96(9):1061–1081, 2016.
- [276] L. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM journal on scientific and statistical computing*, 4(1):136–148, 1983.
- [277] A.C. Hindmarsh. ODEPACK, a systematized collection of ODE solvers. *Transactions on Scientific Computation*, 1:55–64, 1983.
- [278] K Radhakrishnan and A.C. Hindmarsh. Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations. Technical Report UCRL-ID-113855, Lawrence Livermore National Lab, 12 1993.
- [279] L.R. Petzold. Description of DASSL: a differential/algebraic system solver. Technical Report SAND-82-8637, Sandia National Labs, 1982.
- [280] J. Kraft, T. Meyer, and B. Schweizer. Parallel co-simulation approach with macro-step size and order control algorithm. In *15th International Conference on Multibody Systems, Nonlinear Dynamics, and Control*, volume 6, 08 2019.
- [281] M. Zhuang and W. Mathis. Research on stepsize control in the bdf method for solving differential-algebraic equations. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, volume 5, pages 229–232. IEEE, 1994.
- [282] J.W. Weber, K.C. Tang, and M.Q. Brewster. Ignition of composite solid propellants: model development, experiments, and validation. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 4629, 2003.

- [283] V. Carlo. Simulations de transferts thermiques aux parois de moteurs à propergol solide. Technical report, ONERA, 2017.
- [284] W.J. Parker, R.J. Jenkins, C.P. Butler, and G.L. Abbott. Flash method of determining thermal diffusivity, heat capacity, and thermal conductivity. *Journal of Applied Physics*, 32(9):1679–1684, 1961.
- [285] D. Hanson-Parr and T.P. Parr. Thermal properties measurements of solid rocket propellant oxidizers and binder materials as a function of temperature. *Journal of Energetic Materials*, 17:1–48, 1999.
- [286] R.R. Willey, R.W. George, J.G. Ohmart, and J.W. Walvoord. Total reflectance properties of certain black coatings (from 0.2 to 20.0 micrometers). In *Generation, Measurement and Control of Stray Radiation III*, volume 384, pages 19–26. International Society for Optics and Photonics, 1983.
- [287] R. Frederick, J.C. Traineau, and M. Popo. Review of ultrasonic technique for steady state burning rate measurements. In *36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2000.
- [288] J.J. Murphy and H. Krier. Evaluation of ultrasound technique for solid-propellant burning-rate response measurements. *Journal of propulsion and power*, 18(3):641–651, 2002.
- [289] F. Cauty, Y. Fabignon, and C. Erades. Radiative ignition of solid propellants: A practical approach. *International Journal of Energetic Materials and Chemical Propulsion*, 9:285–304, 01 2010.
- [290] F.E. Culick. T-burner testing of metallized solid propellants. Technical report, California Institute of Technology, 1974.
- [291] M.Q. Brewster, M.H. Hites, and S.F. Son. Dynamic burning rate measurements of metalized composite propellants using the laser-recoil technique. *Combustion and flame*, 94(1-2):178–190, 1993.
- [292] J. Traineau, M. Prevost, and P. Tarrin. Experimental low and medium frequency determination of solid propellants pressure-coupled response function. In *30th Joint Propulsion Conference and Exhibit*, 1994.
- [293] B. Bourasseau. Code COPPELIA: version 3.0. calcul de la composition de produits de combustion contenant de nombreuses phases condensées. ONERA Report, RT 40/4386 DEFA/N.
- [294] G. Lengellé, J. Duterque, and J.F. Trubert. Combustion of solid propellants. Technical Report RTO-EN-023, ONERA, 2002.
- [295] P. Boggs and J. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 01 1995.
- [296] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.
- [297] V. Eglajs and P. Audze. New approach to the design of multifactor experiments. *Problems of Dynamics and Strengths*, 35(1):104–107, 1977.
- [298] S. Wang. Étude 1d de la dynamique des flammes à l’aide de méthodes numériques d’ordre élevé. Technical report, Centrale Supélec - EM2C, 2021.
- [299] I.S. Tseng and V. Yang. Combustion of a double-base homogeneous propellant in a rocket motor. *Combustion and Flame*, 96(4):325–342, 1994.

- [300] T.S. Roh, I.S. Tseng, and V. Yang. Effects of acoustic oscillations on flame dynamics of homogeneous propellants in rocket motors. *Journal of Propulsion and Power*, 11(4):640–650, 1995.
- [301] T.S. Roh, S. Apte, and V. Yang. Transient combustion response of homogeneous solid propellant to acoustic oscillations in a rocket motor. *Symposium (International) on Combustion*, 27(2):2335–2341, 1998.
- [302] S. Apte and V. Yang. Unsteady flow evolution and combustion dynamics of homogeneous solid propellant in a rocket motor. *Combustion and Flame*, 131(1-2):110–131, 2002.
- [303] N. Lupoglazoff and V. Francois. Simulation numérique bidimensionnelle des écoulements instationnaires dans les propulseurs à propergol solide. *La Recherche Aéronautique*, 2:21–41, 1992.
- [304] E. Daniel, N. Lupoglazoff, F. Vuillot, T. Basset, and J. Dupays. 2D Navier-Stokes stability computations for solid rocket motors - Rotational, combustion and two-phase flow effects. In *33rd Joint Propulsion Conference and Exhibit*, 1997.
- [305] J. Bur. Étude des transferts radiatifs dans les propulseurs à propergol solide, stage MR2. Master’s thesis, Mines de Paris, 2002.
- [306] C. Geuzaine and J.F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [307] I.S. Tseng and V. Yang. Combustion of a double-base homogeneous propellant in a rocket motor. *Combustion and Flame*, 96(4):325–342, 1994.
- [308] A.I. Volpert, V. Volpert, and V.A. Volpert. *Traveling wave solutions of parabolic systems*, volume 140. American Mathematical Society, 1994.
- [309] C.M. Brauner, A. Lunard, and C. Schmidt-Lainé. Stability of travelling waves with interface conditions. *nonlinear Analysis, Theory and Applications*, 19(5):455–474, 1992.
- [310] C.M. Brauner, L. Lorenzi, and M. Zhang. Stability analysis and Hopf bifurcation at high Lewis number in a combustion model with free interface. *Annales de l’Institut Henri Poincaré C, Analyse non linéaire*, 37(3):581–604, 2020.
- [311] S.B. Margolis and F.A. Williams. Diffusional/thermal instability of a solid propellant flame. *SIAM Journal on Applied Mathematics*, 49(5):1390–1420, 1989.
- [312] P.J. van der Houwen and B.P. Sommeijer. Iterated Runge-Kutta methods on parallel computers. *SIAM journal on scientific and statistical computing*, 12(5):1000–1028, 1991.
- [313] P.J. van der Houwen, B.P. Sommeijer, and W. Couzy. Embedded diagonally implicit Runge-Kutta algorithms on parallel computers. *Mathematics of Computation*, 58(197):135–159, 1992.
- [314] P.J. Van der Houwen and J.J.B. De Swart. Triangularly implicit iteration methods for ODE-IVP solvers. *SIAM Journal on Scientific Computing*, 18(1):41–55, 1997.
- [315] C.A. Kennedy and M.H. Carpenter. *Diagonally Implicit Runge-Kutta methods for ordinary differential equations, a review*. National Aeronautics and Space Administration, Langley Research Center, 2016.
- [316] S. Sadjina and E. Pedersen. Energy conservation and coupling error reduction in non-iterative co-simulations. *Engineering with Computers*, 36(4):1579–1587, 2020.

- [317] S. Sicklinger. *Stabilized Co-Simulation of Coupled Problems Including Fields and Signals*. PhD thesis, Technische Universität München, 2014.
- [318] P. Li, T. Meyer, D. Lu, and B. Schweizer. Numerical stability of explicit and implicit co-simulation methods. In *COUPLED VII: proceedings of the VII International Conference on Computational Methods for Coupled Problems in Science and Engineering*, pages 1249–1260. CIMNE, 2017.
- [319] J. Kraft, S. Klimmek, T. Meyer, and B. Schweizer. Implicit co-simulation and solver-coupling: Efficient calculation of interface-jacobian and coupling sensitivities/gradients. *Journal of Computational and Nonlinear Dynamics*, 2021.







**Titre:** Modélisation et simulation multiphysiques de l'allumage d'un moteur à propergol solide

**Mots clés:** propergols solides, allumage, DAE, couplage

### Résumé:

Les moteurs à propergols solides (MPS) utilisent la combustion d'un matériau solide, le propergol, comme source d'énergie. Une étape cruciale du fonctionnement d'un tel moteur est le transitoire d'allumage, qui peut être décomposé en une phase d'initiation obtenue par un dispositif pyrotechnique, suivie d'une phase de propagation de la flamme le long du chargement de propergol.

L'allumage d'un MPS fait intervenir une grande variété de phénomènes, parmi lesquels : transfert de chaleur conjugué entre les gaz d'allumeur et le propergol, pyrolyse du propergol sous la surface, dégagement d'espèces gazeuses qui brûlent en une flamme intense attachée à la surface, échauffement du propergol par le rayonnement émis par la phase gaz, écoulement multiphasique compressible dans la chambre de combustion, écoulement supersonique dans la tuyère après rupture de l'opercule. Ce caractère multiphysique et les disparités d'échelles spatio-temporelles associées rendent impossible la simulation de l'allumage à l'aide d'un seul outil qui inclurait une modélisation complète de tous les phénomènes. Typiquement, la flamme du propergol est si fine qu'elle ne peut être raisonnablement résolue dans un maillage CFD pour un moteur complet. De plus, elle fait intervenir des cinétiques chimiques raides et potentiellement complexes.

C'est pourquoi l'approche classique est d'utiliser un modèle 1D de la combustion du propergol, en chaque facette limite du domaine CFD correspondant à la surface du propergol. Ainsi, toute la complexité physico-chimique et numérique de la résolution de cette combustion est contenue dans une condition aux limites dynamique. Cependant, les modèles 1D existants sont très simplifiés, ce qui impacte la fidélité de la reproduction de l'allumage dans certains moteurs.

Nous choisissons dans cette thèse d'utiliser une approche 1D plus évoluée, incluant une résolution numérique de la flamme, capa-

ble d'utiliser des cinétiques complexes ou globales. Une attention spécifique est portée à l'analyse mathématique du modèle 1D en stationnaire, à travers l'étude d'une onde progressive de combustion, permettant de clarifier la notion de valeur propre pour la vitesse de régression. Afin de résoudre la combustion en instationnaire, une semi-discrétisation en espace est obtenue par la méthode des lignes. La nature différentielle-algébrique du système d'équations obtenu est clairement exposée, ce qui permet de choisir des méthodes d'intégration performantes pour résoudre la dynamique du propergol avec un ordre élevé en temps et un pas de temps adaptatif.

Afin d'assurer une bonne reproduction de l'allumage de différents propergols, un processus d'optimisation est mis en place afin de paramétrer automatiquement le modèle pour reproduire les temps d'allumage observés expérimentalement.

Le modèle 1D est ensuite couplé avec le code CFD 3D CEDRE de l'ONERA, afin de permettre la simulation de l'allumage de moteurs complets. Le couplage est initialement opéré à l'ordre 1, mais des techniques sont suggérées pour permettre un ordre élevé et un couplage adaptatif.

Afin de vérifier l'effet de la représentation 1D de la flamme, un couplage plus détaillé est aussi développé, où la flamme est résolue dans le code CFD lui-même. La comparaison des 2 approches sur une configuration académique en 2D permet de vérifier la cohérence et la précision de la nouvelle approche.

Le couplage entre le modèle 1D et le code CFD développé durant cette thèse et les approches interdisciplinaires utilisées offrent de nouvelles perspectives tant pour le développement d'outils mathématiques pour des stratégies de couplage adaptatives pour une large gamme applications, permettant d'optimiser la précision et le coût des calculs, que pour une meilleure prévision ou restitution de l'allumage de différents moteurs.

**Title:** Multiphysical modelling and simulation of the ignition transient of complete solid rockets motors

**Keywords:** solid rocket motor, solid propellant, ignition, DAE, coupling

### Abstract:

Solid rocket motors (SRMs) use the combustion of a solid material, the propellant, as an energy source. A crucial step in the operation of such an engine is its ignition, during which the surface of the propellant must be heated by about 400 degrees to initiate combustion. This is usually done by letting a hot jet impact the surface.

The ignition of an SRM involves a wide variety of phenomena, including: combined heat transfer between the igniter gases and the propellant, pyrolysis of the propellant below the surface, release of gaseous species that burn in an intense flame attached to the surface, heating of the propellant by radiation emitted from the gas phase, compressible multiphase flow in the combustion chamber, supersonic flow in the nozzle. The multiphysical nature and the disparities in space-time scales make it impossible to simulate ignition using a single tool that would include a complete modelling of all the phenomena. Typically, the propellant flame is so thin that it cannot be reasonably resolved in a CFD mesh for a complete motor. In addition, it involves stiff and potentially complex chemical kinetics. This is why the classical approach is to use a 1D model of the propellant combustion, at each boundary face of the CFD mesh belonging to the propellant surface. Thus, all the physico-chemical and numerical complexity of solving this combustion is encapsulated in a dynamic boundary condition. However, the existing 1D models are very simplified, impacting the fidelity of the reproduction of ignition in some motors.

In this thesis, we choose to use a more advanced 1D approach, which includes a numerical resolution of the flame, able to use complex or global kinetics. Specific attention is paid to the mathematical

analysis of the 1D model in steady state, through the study of a travelling combustion wave, clarifying the notion of eigenvalue for the regression speed. To simulate unsteady combustion, a semi-discretisation in space is obtained by the method of lines. The differential-algebraic nature of the resulting system of equations is clearly exposed, allowing for the choice of efficient integration methods to solve the propellant dynamics with high order in time and adaptive time step.

In order to ensure an accurate reproduction of the ignition of different propellants, an optimisation process is developed to automatically parameterise the model, allowing for a good agreement between experimental and simulated ignition times.

The 1D model is then coupled with the semi-industrial 3D CFD suite CEDRE from ONERA, in order to allow for the simulation of ignition in complete motors. The coupling is initially operated at order 1, but techniques are suggested to allow for a higher-order and adaptive coupling.

In order to verify the effect of the 1D representation of the flame, a more detailed but more expensive coupling is also developed, where the flame is solved in the CFD code itself. The comparison of the two approaches on an academic 2D configuration allows to verify the consistency and accuracy of the new approach.

The coupling between the 1D model and the CFD code developed during this thesis and the interdisciplinary approaches used offer new perspectives both for the development of mathematical tools for adaptive coupling strategies for a wide range of applications, allowing to optimise the accuracy and the cost of the computations, as well as for a better reproduction of ignition in various motors.