Model reduction for thermal management of high power electronic components for aerospace application

Hazem Ben Aissia

N°d'ordre NNT : 2019LYSEI071

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
(INSA LYON)

Ecole Doctorale ED162
MEGA de Lyon
(Mécanique, Energétique, Génie Civil, Acoustique)

Spécialité de doctorat : Thermique et Energétique

Soutenue publiquement le 25/9/2019, par :
Hazem Ben Aissia

Model reduction for thermal management of high power electronic components for aerospace application

---

Devant le jury composé de :

Bataille, Françoise	Professeur, Université de Perpignan	Présidente
Podvin, Bérengère	Chargé de recherche HDR (LIMSI, CNRS)	Rapporteure
Videcoq, Etienne	MCF HDR (ENSMA Poitiers)	Rapporteur
Laraqi, Najib	Professeur (Université Paris Nanterre)	Examinateur
Giroux Julien, Stéphanie	MCF HDR (Université Claude Bernard Lyon 1)	Examinatrice
Rouizi, Yassine	MCF (Université d'Evry Val d'Essonne)	Examinateur
Xin, Shihe	Professeur (INSA-Lyon)	Directeur de thèse
Jay, Jacques	Professeur (INSA-Lyon)	Directeur de thèse
Knikker, Ronnie	MCF (INSA-Lyon)	Invité

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage **secretariat@edchimie-lyon.fr** INSA : R. GOURDON	M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec. : M.C. HAVGOUDOUKIAN **ecole-doctorale.eea@ec-lyon.fr**	M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 **gerard.scorletti@ec-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : H. CHARLES **secretariat.e2m2@univ-lyon1.fr**	M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://www.ediss-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE **secretariat.ediss@univ-lyon1.fr**	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 **emmanuelle.canet@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 **infomaths@univ-lyon1.fr**	M. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 **zamboni@maths.univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction **ed.materiaux@insa-lyon.fr**	M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 **jean-yves.buffiere@insa-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction **mega@insa-lyon.fr**	M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **http://ed483.univ-lyon2.fr** Sec. : Viviane POLSINELLI Brigitte DUBOIS INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 **viviane.polsinelli@univ-lyon2.fr**	M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Contents

List of Figures

List of Tables

Nomenclature

Roman symbols

$\dot{m}$	Mass flow rate (Kg/s)
$\mathbf{a}(t)$	Temporal amplitude vector
$\mathbf{C}$	Heat capacitance (mass) matrix in FEM
$\mathbf{K}$	Heat conductance (stiffness) matrix in FEM
$\mathbf{l}$	Load vector (FEM)
$\mathbf{p}$	parameters vector
$\mathbf{S}$	Selection matrix
$\mathbf{U}$	Snapshots matrix
$\mathbf{u}(t)$	Snapshot vector
$\mathbf{x} = (x, y, z)$	Space coordinates vector
A	Area (m^2)
C_p	Heat capacity $(J/K/Kg)$
D	Duty cycle or Diameter
E	Energy (J)
E	Error
h	convection heat coefficient $(W/m^2/K)$
I	The current (A)
k	Thermal conductivity (W/m/K)
L	Length
M	Number of nodes
P	Electric power (W) or Perimeter (m)
q	Heat flux density (W/m^2)
R	Electric reistance (Ω)
r	Number of retained modes
R_{th}	Thermal resistance (K/W)
T	Temperature (K)
t	Time (s)
U	The voltage (V)

Greek symbols

$\mathbf{\Lambda}$	POD Eigenvalues matrix
$\mathbf{\Phi}$	POD basis
$\mathbf{\Sigma}$	SVD Singular values matrix
Δ	Variation
ϵ	Emissivity coefficient
η	Efficiency
γ	Mechanical strain
λ	Failure rate
λ	POD eigen value
μ	Dynamic viscosity $(Kg/m/s)$
ϕ	Heat flux (W)
σ	Singular value

Subscripts

a	Activation
$cond$	Conduction
cv	Convection
D	Drain (of the MOSFET)
DS	Drain to Source

eff Effective

eq Equivalent

f Failure or Fluid or Fin

GS Gate to Source

h Hydraulic

j Junction

kn Known

m Mean

off Off state

on On state

p Wall

Q Power losses

s Solid

sur Surrounding

sw Switching

th Threshold

$unkn$ Unknown

Superscripts

HS Heat Sink

in Inlet

out Outlet

Abbreviation

CTE Coefficient of Thermal Expansion

FEM Finite Element Model

$I2MPECT$ Intelligent Integrated Modular Power Electronic Conveter

$IROM$ Inverse Reduced Order Model

LHP Loop Heat Pipe

MEA More Electric Aircraft

$MOSFET$ Metal Oxide Semiconductor Field Effect Transistor

PDE Partial Differential Equation

PEM Power Electronic Module

POD Proper Orthogonal Decomposition

REC Relative Energy Content

RMS Root Mean Square

ROM Reduced Order Model

SVD Singular Value Decomposition

Operators

$(.)^{\dagger}$ Pseudo Inverse of a matrix

$\nabla()$ Gradient

$\nabla.()$ The divergence

À mes parents

Remerciements

Je voudrais tout d'abord remercier Madame Bérengère Podvin et Monsieur Etienne Videcoq d'avoir rapporté ma thèse ainsi que pour leurs remarques pertinentes sur mon travail. Je remercie également Monsieur Najib Laraqi, Monsieur Yassine Rouizi, Madame Stéphanie Julien Giroux et Madame Françoise Bataille d'avoir examiné mon manuscrit de thèse et avoir participé à mon jury de thèse ainsi que pour la qualité de l'échange entretenue au cours de la soutenance.

Je commence par remercier vivement mes directeurs de thèse les professeurs Shihe Xin et Jacques Jay, de m'avoir accueilli dans l'équipe thermique du laboratoire (CETHIL) de l'INSA de Lyon. Je remercie Shihe Xin pour sa confiance, ses conseils scientifiques, et ses remarques pertinentes. Je voudrais exprimer ma gratitude à Jacques d'avoir suivi tout le travail, pour tout le temps gigantesque consacré pour encadrer cette thèse, pour ses remarques scientifiques pertinentes. Merci Jacques de m'avoir épaulé pendant les moments difficiles de cette thèse. Je remercie Ronnie Knikker pour son encadrement, ses remarques et ses conseils.

Je remercie tous les membres du consortium européen I2MPECT pour toutes les discussions techniques très intéressantes et de très haut niveau.

Je voudrais exprimer mes sincères remerciements au professeur Bruno Allard manager du projet I2MPECT, ainsi qu'aux membres du laboratoire Ampère (Guy, Christian, Malorie, Ossaynou), de m'avoir impliqué dans leurs discussions sur la fiabilité des composants électroniques et de l'intérêt qu'ils ont porté à mon travail.

Mes remerciements les plus vifs à tous les doctorants du laboratoire CETHIL pour la bonne ambiance ainsi que pour les moments conviviaux partagés ensemble : Sébastien, Damien, Francesco, Nicolas, Momath, Ghady, Loic, Martin, Yassine, Karolina, Moomal, Naveed, Kai, Bao, Sandra, Chi Kien, Thibault, Quentin, Ali, Khristia, Eloise, Rémi, Mathilde, Jean, Samuel, Sophie, Sofiane, Fouad, Nabil, Christophe, Aurélia, Etienne, … Merci Les amis !

Je remercie tous les autres membres du labo administratif, chercheurs, professeurs et personnels de l'atelier: Pierre Olivier, Serge, Valérie, Anthony, Xavier, Nicolas, Florence, …

Une pensée particulière va à l'équipe du foot du CASI pour les matches de foot hebdomadaires qui nous ont permis de se renouveler l'énergie : Etienne, Jo, Umir, Kamel, POC, Momo, Michel, Jeremy, Nico, Antonio, Thomas, les deux Christian, Pierre, Patrick, Fabien, Aurélien, Muller, Mahmoud, Mickael, Basile, … Merci les amis !

Je remercie mes amis Lyonnais et ma deuxième famille avec qui j'ai passé des bons moments pendant la période de mes études ingénieurs à Lyon : Pierre Alain, Bruno et Robin. Une pensée particulière va à Pierre Alain et tous les weekends passés ensemble, Merci Pierro d'être un ami si particulier! Merci Robin pour tous les weekends passés à Carpentras au cours desquels les matches de pétanque finissent après minuit, Merci BruBru pour les soirées crêpe et ciné. Mes remerciements vont aussi à mes amis Hamza et Oussama pour tous les moments amicaux, les repas partagés ensemble.

Mes remerciements vont aussi vers tous les professeurs que j'ai eus depuis ma première

année scolaire pour le savoir et les valeurs qu'ils nous ont appris.

Je tiens à remercier aussi la famille Mokrani : Tata Naziha, Sami, ''Monsieur le professeur'' Fadhel, Tata Ons, Tata Ahlem, pour votre support et soutien pendant mes années d'étude en France ainsi qu'au cours de ma thèse.

Je remercie toute ma famille en Tunisie. Une pensée particulière va à mes deux frères Nafaa et Hakim, à ma tante Rawdha, Majda (mani nasi lifriki fi inaya) et mon oncle Noureddine.

Ce dernier remerciement va aux personnes les plus chères : mes parents. Merci pour tous vos sacrifices afin de nous donner avec mes frères les meilleures conditions de vie et d'éducation. Ce travail de thèse n'aurait jamais vu la lumière sans votre encouragement. Les remerciements n'arriveront jamais à décrire ma reconnaissance envers vous.

Introduction

Industrial context

The transition to electric vehicles is one of nowadays big challenges. Electric vehicles are less heavy, less noisy and simple to diagnose. In addition, they have reduced carbon footprint on the environment. These reasons encourage transport companies to invest in the more electric aircraft (MEA) idea [1].

In MEA, we use the electrical power for extracting and distributing the non-propulsive powers. The electrical power comes from the aircraft main engine, which operates using combustion process. The non-propulsive aircraft systems can be wing anti-icing, environmental systems (ECS), flight control actuators, landing gear brakes, avionics lighting, fuel and oil pumps, etc. Figure 1 shows the difference between a conventional aircraft and a MEA.

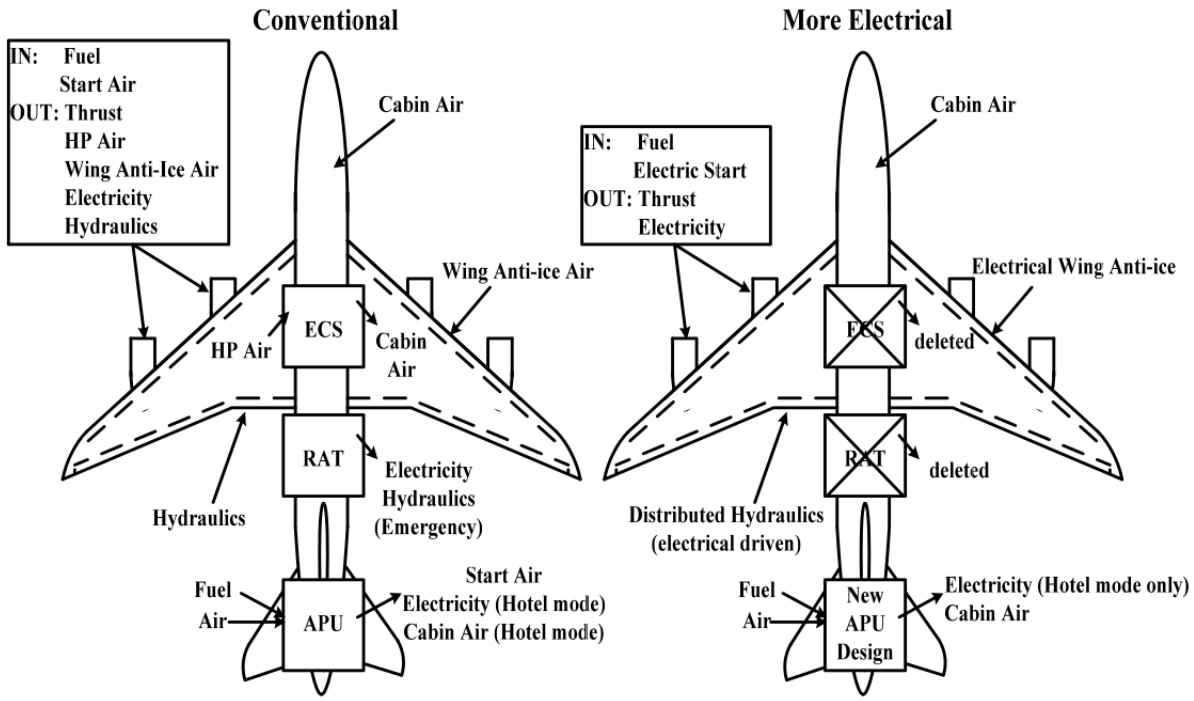


Figure 1 – Comparison between conventional aircraft systems and MEA systems [1]

The main reasons behind the use of electrical systems is to decrease operating and maintenance costs, improve the power system efficiency and reliability [2].

Actuators on board of MEA do not require the same form (Direct Current or Alternating Current) and amount of electric energy, that is why we use electronic converters that adjust the

need of each actuator.

For this reason, a consortium of European industrial and academic partners launched the I2MPECT (Integrated, Intelligent modular power electronic converter) project.

The I2MPECT project aims to design and manufacture a high power DC/AC converter intended for MEA. Partners of this project are listed in Figure 2. The CETHIL lab (INSA Lyon) as an academic partner is in charge of the design of the cooling system and the creation of a compact thermal model for the electronic package (electronic converter and the cooling system).

Figure 2 – The I2MPECT project consortium

Thesis objective

The main drawback of electric systems is that as increasing the power, the temperature of electronic components increases and this results in a decrease of the efficiency and the lifetime. Accordingly, a monitoring strategy is needful to predict the chips temperature and apply preventive action before failure.

Many methods exist to monitor the chips temperature: direct measurement of the temperature, indirect measurement using Thermo-Sensitive Electrical Parameters (TSEP) or by thermal models.

Direct measurement consists in placing a sensor on the electronic components. This solution is discarded for packaging reasons. Otherwise, temperature sensors have been used on the electronic card but are located far from chips. Thus, the measured temperature is different from the chips temperature. In this situation, the monitoring will be carried out using a thermal model. Such a model should be accurate in order to give a reliable information on the electronic component health, and also compact in order to compute in real time the temperature. To fulfill this need, the reduced order model can be used.

In practical situations, boundary conditions and power losses in electronic components can be known or unknown. When they are known, a classic reduced order model based on proper orthogonal decomposition (POD) and Galerkin projection can be used. The latter model will be called the direct reduced order model (DROM). When they are unknown, an inverse reduced order model (IROM) will be employed and is based on proper orthogonal decomposition and a minimization procedure. In the latter model, the only input is the information of the temperature sensors.

The aim of this thesis is to develop these two models.

Thesis outline

This thesis is divided into five chapters. The content of each chapter is discribed as follows:

- In the first chapter, a literature review on the relationship between the reliability and the temperature is undertaken. The existing cooling solutions are described as well as their

adavantages and drawbacks. The cooling solution respecting the technical specifications of the project is presented.

- Chapter 2 provides a description of the finite element modeling of heat transfer in the power module. After introducing the heat transfer governing equations, its finite element discretization is adressed. This method is formulated in the case of solid medium, fluid medium and conjugate heat transfer (fluid and solid medium in contact).
- Chapter 3 gives a review on model reduction methods. The POD method is described in details. The usage stage for the reduced order model based on Galerkin projection is formulated in the case of diffusion equation. The usage stage for the inverse reduced order model is adressed in details.
- In Chapter 4, an application of the reduced order model based on POD and Galerkin projection is given. The application treats one leg power module in which the cooling system is modeled by a heat transfer coefficient. The performances of the ROM in terms of accuracy and computing time are investigated.
- Chapter 5 presents an application of the IROM in the case of the entire system. In this application the conjugate heat transfer within the electronic package is modeled. The robustness of the IROM to deal with various operating conditions by knowing only the temperature sensors is adressed.

Bibliography

[1] AA AbdElhafez and AJ Forsyth. A review of more-electric aircraft. In *13th International Conference on Aerospace Science & Aviation Technology (ASAT-13), Paper No. ASAT-13-EP-01*, 2009.

[2] Jeremy Bourdon, Pascal Asfaux, and Alvaro Morentin Etayo. Review of power electronics opportunities to integrate in the more electrical aircraft. In *Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS), 2015 International Conference on*, pages 1–6. IEEE, 2015.

Chapter 1

State of the art of thermal management of high power electronic components and the studied system

Failure occuring in electronic packages are mainly due to excessive temperature amplitude and cycling. In order to improve the reliability of electronic devices and increase its lifetime, a control strategy of its temperature should be undertaken. Therefore, a cooling strategy is required. The aim of this chapter is to give an overview of thermal management solutions for electronic components and to choose the suitable solution when taking into consideration technical specifications. The first section of this chapter will describe the studied electronic power module as well as power losses occuring in semi-conductor devices. In this section the reliability of electronic components is briefly depicted and some existing lifetime predictive models are introduced. The third section gives a non extensive literature review of existing cooling solutions. The chosen cooling solution is presented and modeled after introducing technical specifications.

Sommaire

1.1 Power electronic converter

In this section, the studied power electronic module will be described. Power loss mechanisms in a single MOSFET will be depicted as well as a method to calculate it. Then, the reliability of electronic components is described in a generic way.

1.1.1 The I2MPECT project power module description

The I2MPECT consortium aims to conceive and manufacture a three phase $45\ kW$ DC/AC converter. A converter contains two switches that control the electric energy transfer. Switches are generally transistors that can be at the on or the off state. These switches can be MOSFET or IGBT or any other transistor technology. In the I2MPECT project, The technology used is Silicon Carbide (SiC) MOSFETs. This technology offers many advantages. It is known that SiC components are able to reduce the losses in the power module. In addition SiC MOSFETs can be used at high switching frequency (as switching frequency increases, the sine waveform constructed by the converter becomes smooth) compared to other technology (IGBT,JFET,...) ([1,2]). This will result in lower harmonic distortion and thus a loss reduction in electric machines (motor). In picture 1.1, we represent the one leg power converter developed by Siemens. Each leg contains 12 MOSFETs.

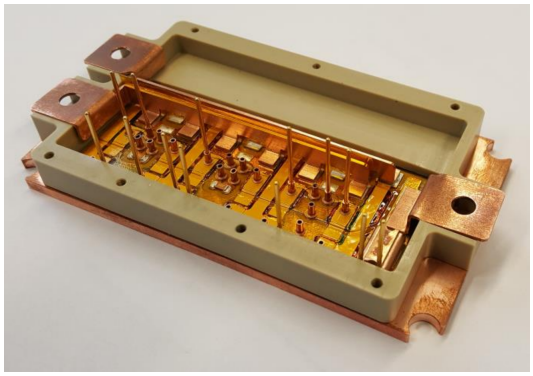


Figure 1.1 – I2mpect one leg power converter

In order to comprehend the thermal behavior of this device, Figure 1.2 represents a cross section of the power module.

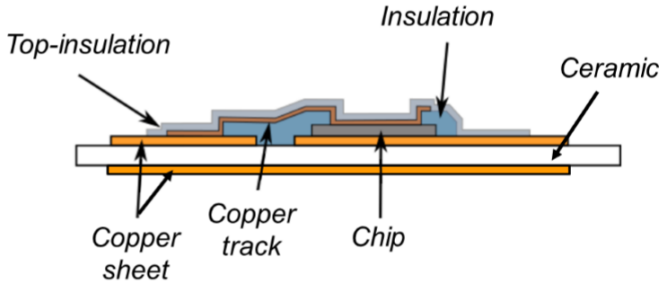


Figure 1.2 – Cross section of the I2mpect Power module

An electric insulator gel covers the chips in order to avoid short circuit and electric arc problems. The MOSFETs are soldered on DBC (Direct Bonded Copper) representing the current

path. The DBC consists of a ceramic tile and a sheet of copper bonded to both sides. The set MOSFETs and DBC are mounted on a copper base plate (not shown in Figure 1.2). The role of the base plate is to spread out the heat on the hot spots created by the dissipated power in MOSFETs.

1.1.2 The power module topology

A DC/AC converter (also called power inverter) is a device able to convert direct current into a sine wave current. This task is accomplished thanks to switches that control the energy transfer between branches of a circuit. We represent on Figure 1.3 the electrical circuit of the 3 phases converter.

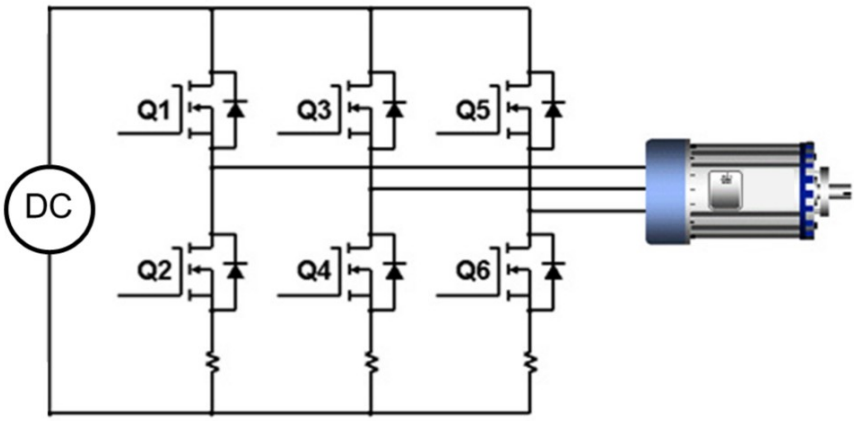


Figure 1.3 – Converter topology

One leg converter contains two switches. A switch can be one or many transistors in parallel. In the case of I2mpect converter, we use six MOSFETs per switch (in parallel) for efficiency reasons. In fact, when increasing the number of MOSFETs per switch the current crossing each MOSFET will be lower and hence the temperature per transistor will decrease.

By turning on and off switches, the converter delivers an output sine waveform. The transition between the on and off state of each MOSFET is driven by a Pulse Width Modulation (PWM) signal entering the gate of each MOSFET. The voltage required to turn on the MOSFET should be greater or equal to a threshold voltage called U_{th}. The condition to set to on state the MOSFET can be written:

$$U_{GS} > U_{th}$$

Where U_{GS} is the voltage between the gate and source (see Figure 1.4). U_{GS} is produced by a device called gate driver.

When increasing the number of MOSFETs per switch, the power density decreases and heat spreading is enhanced. Consequently, the thermal management of the converter becomes easier, but in electrical point of view the task is tough as we have to control the gate of many MOSFETs.

1.1.3 Power losses

The dissipated power in the electronic converter is the sum of conduction and switching losses. These losses are mainly occurring in MOSFETs (we neglect power losses in other components and the DBC).

Conduction losses
Conduction losses are the power dissipated by the Joule effect. These losses are calculated by the formula:

$$P_{cond} = R_{DS(on)} I^2_{D,RMS}$$

Where $R_{DS(on)}$ is the on state resistance of the MOSFET and $I_{D,RMS}$ is the root mean square of the current flowing through the MOSFET. After the transition time of the MOSFET the current reaches a constant value $I_{D(on)}$ and $I_{D,RMS} = I_{D(on)}\sqrt{D}$ where D is the MOSFET duty cycle (see Figure 1.4). The duty cycle D of the MOSFET is the ratio between pulse duration of U_{GS} and the period of the rectangular waveform. Thus, conduction power losses per MOSFET can be written as:

$$P_{cond} = R_{DS(on)} I^2_{D(on)} D$$

When manufacturing two MOSFETs by the same process, they do not have exactly the same $R_{DS(on)}$. The relative difference between the two values can reach 20%. $R_{DS(on)}$ depends on temperature. This implies that computing power losses in a MOSFET requires the knowledge of the temperature and vice versa. Consequently, thermal and electrical calculation should be performed in a coupling loop. This loop defines the electro-thermal model of the electronic device ([3], [4], [5]). The relation giving the temperature variation of $R_{DS(on)}$ depends on the MOSFET model and is generally presented on the manufacturer datasheet. An example of temperature variation of $R_{DS(on)}$ is given in [6].

Switching losses
When the MOSFET is turned on or off, it dissipates power called switching losses. To understand their origin we represent on Figure 1.4 the different waveforms for a single MOSFET when it is turned on and off. The area beneath the power curve during turning on transition period represents the dissipated energy and can be calculated as:

$$E_{on} = I_{D(on)} U_{DD} t_{on} \frac{1}{2}$$

where $t_{on} = t_{fv} + t_{ri}$ is the time for MOSFET to turn on, t_{fv} is the time for voltage to fall and t_{ri} is the time for the current to rise (see Figure 1.4). The switching frequency represents how much time per second the MOSFET turns on and off. Thus switching power losses during the turn on can be obtained by the formula:

$$P_{on} = E_{on} f_{sw}$$

where f_{sw} is the switching frequency. In a similar way, switching losses during turn off can be calculated as:

$$P_{off} = E_{off} f_{sw}$$

where

$$E_{off} = I_{D(off)} U_{DD} t_{off} \frac{1}{2}$$

and t_{off} is the total time for the MOSFET to turn off.

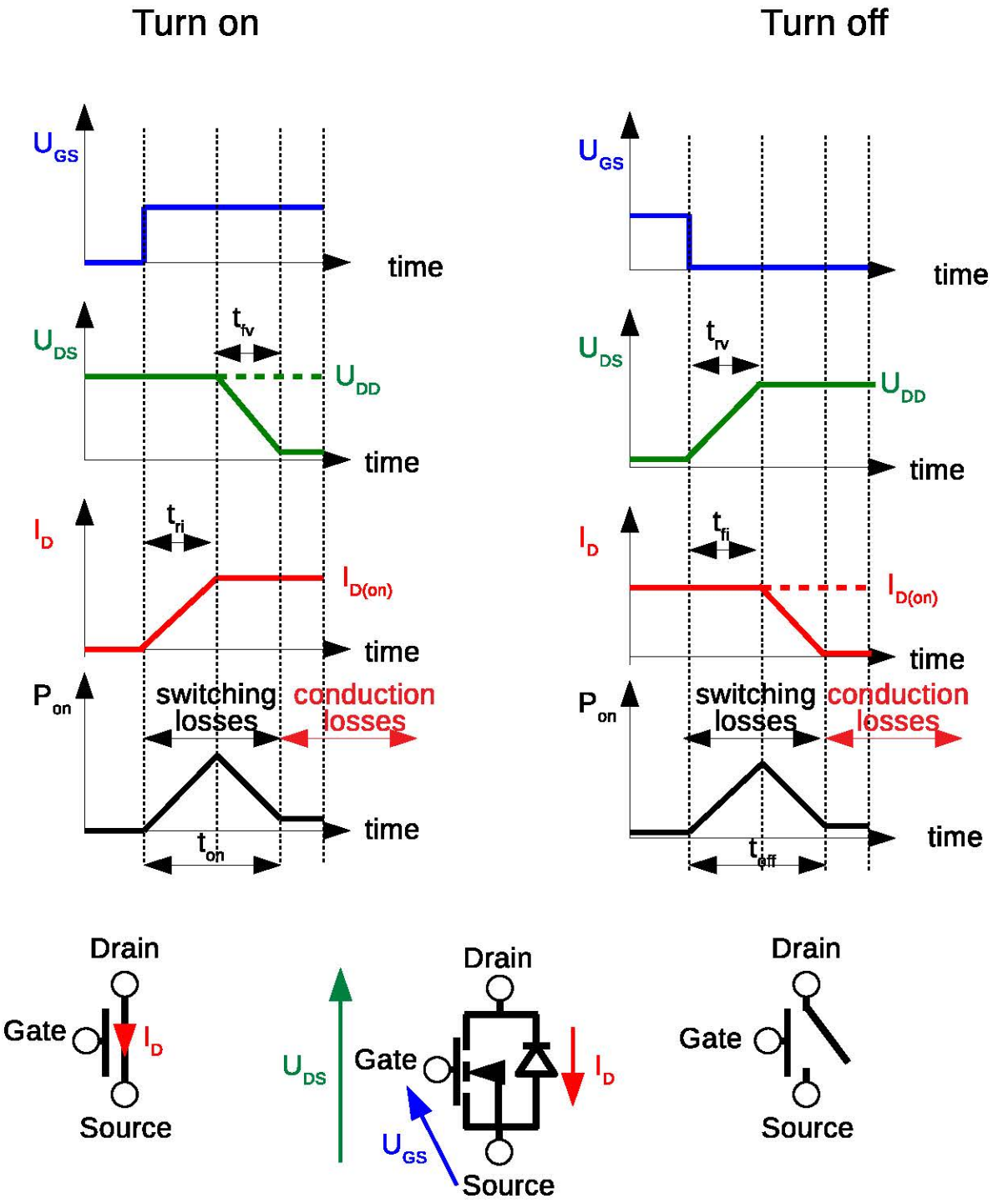


Figure 1.4 – Waveforms in a MOSFET during on and off states

In conclusion, the total losses per MOSFET can be calculated as:

$$P_{total} = P_{sw} + P_{cond}$$

where

$$P_{sw} = P_{on} + P_{off} = f_{sw}(E_{on} + E_{off})$$

and

$$P_{cond} = R_{DS(on)} I^2_{D(on)} D$$

Remark 1. *If the switching frequency increases, the quality of the output sine wave is improved. But as power switching frequency increases, power losses become important and thus the temperature of the electronic component increases. Active thermal control can be used to adjust the optimal switching frequency* [7].

Remark 2. *In reality power losses depend also on the load (motor) and the converter topology. Thus, we should calculate in details the current and the voltage in all branches of the converter in order to estimate accurately power losses. In* [5] *an example of power loss calculation for a complete converter is presented.*

1.1.4 Reliability of electronic devices

1.1.4.1 Reliability description

The reliability analysis of electronic devices is central in defining the physical limits of its operation. Reliability studies aim to identify causes and mechanisms of failures that should be avoided. In general, failures in electronic packaging are due to many factors presented on Figure 1.5.

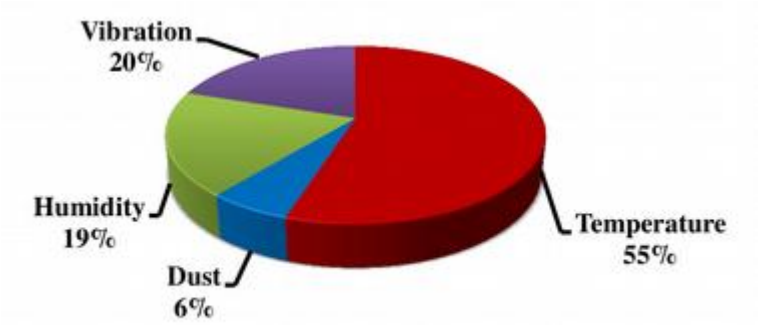


Figure 1.5 – Major Cause in electronic failure [8]

The main electronic failure is due to an important temperature magnitude and cycling. In modern aircrafts, humidity and dust are well controlled. The main concern focus on how to decrease operating temperature and control it.

Failures are mainly created by thermo-mechanical stress cycling. When power electronics heats up, a temperature gradient takes place within the power module. Layers of the power module do not have the same coefficient of thermal expansion (CTE). It is known that the thermo-mechanical strain is proportional to the temperature swing and the CTE. Thus, the layers of power module will expand with different rates. The repeated strain created by thermal cycling will lead to many defects. For instance, we can cite cracks appearing in the solder joint (Figure 1.6).

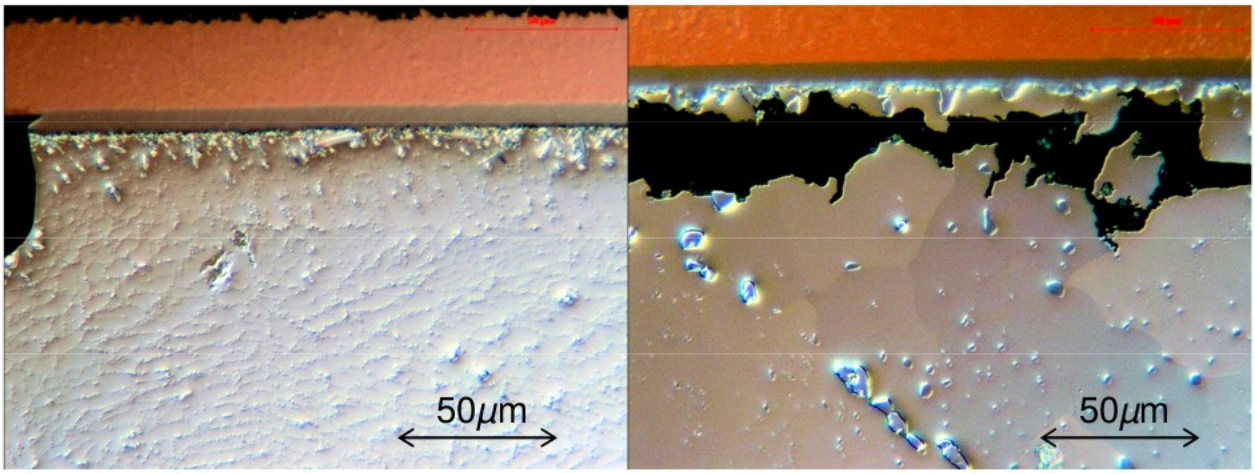


Figure 1.6 – State of a solder ball after manufacturing (left) and after 2257 thermal cycles between $-55°C$ and $125°C$ (right) [9]

As the difference in CTE between layers increases, the probability of occurrence of such defects increases.

The defect appearing at the interface will increase the contact thermal resistance. Hence, the temperature will reach higher values, which will accelerate defect emerging.

1.1.4.2 Reliabilty prediction

Reliability can be seen as the ability of the component to perform a task under given operating condition for a stated period of time [9]. To quantify the reliability of an item, accelerated aging experiments are performed under given condition until it reaches failure. The obtained results allow the calculation of the failure rate $\lambda(t)$ which is the probability of a failure to occur at time t. Details about mathematical tools used to calculate the failure rate function are presented in [10]. The failure rate curve have a bathtub shape presented on Figure 1.7.

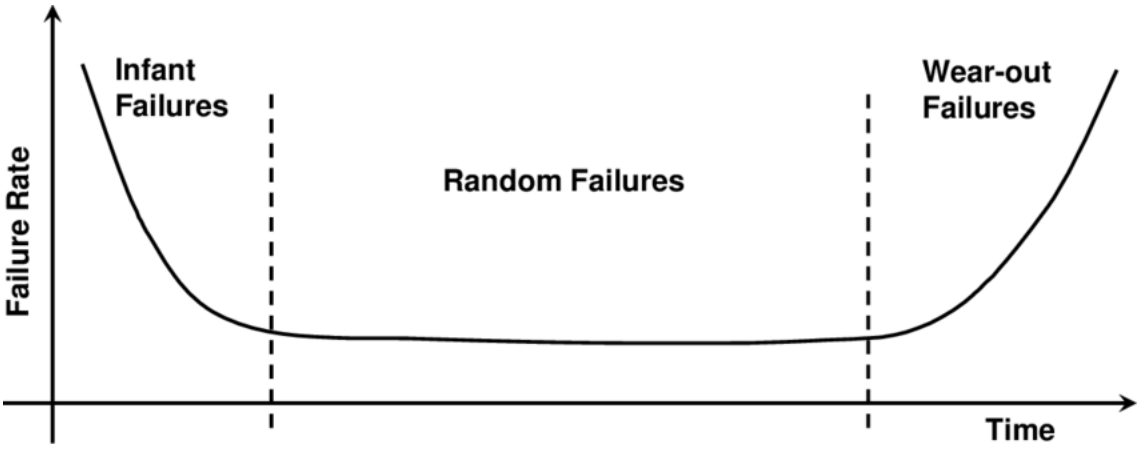


Figure 1.7 – Bathtub curve for electronic device reliability [9]

Figure 1.7 shows that the reliability of an item include three periods :

1. **The burn in or infant mortality period**: the higher failure rate during this period is due to manufacturing imperfections of the item.

2. **The useful life period**: the failure rate during this period is constant. Failure during this period occurs randomly.

3. **the final wear out period**: the dramatic increase of the failure rate illustrates the end of life of the item

Many models are used to quantify the reliability of electronics. The Arrhenius law expresses the failure rate as a function of the steady state temperature of the analysed component. The failure rate can then be written as:

$$\lambda = A \exp(\frac{-E_a}{k_B T})$$

where A is a constant, E_a is the activation energy for the failure mechanism, k_B is Boltzmann's constant and T is the steady state temperature of the electronic component.

The steady state temperature is not the only factor of failure. In general, most failure mechanisms are due to the temperature gradient ($\frac{\partial T}{\partial x_i}$), temperature cycle magnitude (ΔT) and rate of change of temperature ($\frac{\partial T}{\partial t}$) [11].

Another lifetime prediction model based on thermal cycling fatigue is included in Coffin Manson model. This model gives the number of temperature cycles to failure N_f:

$$N_f = \frac{1}{2}\left(\frac{\Delta\gamma}{2\varepsilon_f}\right)^{\frac{1}{c}}$$

where $\Delta\gamma$ is the cyclic strain range. The cyclic strain range depends on the difference in CTE between the component and the PCB, the cyclic temperature extremes, distance from neutral point (commonly the center of the package) and the stand-off height [9]. ε_f is the fatigue ductility coefficient and is equal to 0.325 for standard SnPb solder. c is the fatigue ductility component and is equal to -0.442 for standard SnPb solder. Coffin Manson fatigue law have been experimentally validated for various electronic packaging [12]. Engelmaier modified Coffin Manson law so that the constant c becomes a function of the solder temperature:

$$c = -0.442 - 6\ 10^{-4}\bar{T}_{sj} + 1.74\ 10^{-2}\log(1 + \frac{360}{t_d})$$

where $\bar{T}_{sj}$ is the mean solder joint cycling temperature and t_d is the dwell time (in minutes) at the temperature extremes. Applicability of these models to real geometries is limited due to many physical phenomena that are not taken into account. Calibration of these models for each electronic package should be carried out for trustworthy prediction [9].

1.2 Heat transfer in electronic devices

Whenever a temperature difference exists between two mediums, heat transfer takes place and energy flows from the hot medium to the cold one. Three main modes of heat transfer exit: conduction, convection and radiation. In the following section, we will give a non-extensive description of these modes and see how they are involved in the cooling of electronic components.

1.2.1 Heat transfer modes

1.2.1.1 Conduction

Heat conduction is the energy transfer across a medium (solid, liquid or gases). The physical mechanism on which relies this mode is the random atomic and molecular activity within and through a medium. Conduction can be viewed as the transfer of energy from the more energetic to the less energetic particles of a substance due to interaction between the particles ([13]).

The phenomenological Fourier's law governs conductive heat transfer. This law can be written as:

$$\mathbf{q}(\mathbf{x},t) = -k(\mathbf{x},t)\ \nabla T(\mathbf{x},t) \tag{1.1}$$

where $\mathbf{q}$ is the local heat flux density (W/m^2), k is the medium thermal conductivity $(W/m/K)$ and ∇T is the temperature gradient (K/m).

The sign minus in front of thermal conductivity illustrates the fact that heat flows from the hotter medium to the colder one. This fact is in agreement with the second law of thermodynamics.

1.2.1.2 Convection

Heat convection occurs when a moving fluid is in contact with a surface and that the two are at different temperatures. This transfer mode is created by the contribution of two mechanisms: the random molecular motion(diffusion) and energy transferred by bulk motion of the fluid (advection).

Convection heat transfer is governed by Newton's law. This law does not focus on details of the heat transfer process and can be written as:

$$q(\mathbf{x},t) = h(\mathbf{x},t)\ \ (T_s(\mathbf{x},t) - T_f) \tag{1.2}$$

where q is the convective heat flux (W/m^2), T_s is the solid surface temperature, T_f is the fluid temperature far away from the solid and h is the convection heat transfer coefficient $(W/m^2/K)$.

In practice h is obtained from a dimensionless number called Nusselt number (Nu) that can be determined either by numerical simulation or by experimental method. The Nusselt number Nu is defined as :

$$Nu = \frac{h.L_c}{k_f}$$

where L_c is a characteristic length and k_f is the fluid thermal conductivity. Determination of this coefficient is the main challenge in convection heat transfer studies. This coefficient depends mainly on fluid velocity and thermal properties as well as the solid shape. When choosing the convection heat transfer correlation (Nusselt number), a particular focus should be made on the following points:

1. The nature of the flow (internal or external flow).
2. The flow regime (laminar or turbulent flow). This point is checked by calculating Reynolds number and comparing it to the critical one.
3. The solid shape and roughness (at the contact surface with the fluid).
4. The fluid thermodynamic and transport properties.
5. The cause of fluid motion (natural or forced convection).
6. The flow nature (single or two phase flow).
7. The kind of the boundary condition at the solid fluid interface (uniform temperature or uniform heat flux).
8. The development of the flow (hydrodynamically or thermally or both developed flow). This point is checked by calculating the hydrodynamic and thermal entry length. In laminar flow the hydrodynamic entry length L_{hyd} can be obtained from the formula [13]:

$$\frac{L_{hyd}}{D_h} = 0.05\ Re_{D_h}$$

where D_h is the hydraulic diameter and Re is the Reynolds number.

Typical values of heat transfer coefficient are presented in Table 1.1:

Process	$h\ (W/m^2/K)$
Free convection	
Gases	2-25
Liquids	50-1000
Forced convection	
Gases	25-250
Liquids	100-20000
Convection with phase change	
Boiling or condensation	2500-100000

Table 1.1 – Typical values of convection heat transfer coefficient [13]

When choosing single phase cooling the best option is to operate in forced convection with liquid fluid as it can be seen from Table 1.1. The best convection cooling option is to use convection with phase change. Two phase cooling devices will be described in section 1.3.4 and its operating limits will be outlined. The chosen convection cooling option should fit industrial specifications.

1.2.1.3 Radiation

Thermal radiation is the only heat transfer mode that does not require a physical medium. Heat transfer by radiation occurs between two surfaces having different temperature and that are not necessarily in contact or separated by matter. The physical mechanism governing thermal radiation is the electromagnatic waves emitted by matter as result of the change in the electronic

configurations of the atoms or molecules [14]. The thermal radiation energy emitted by a surface can be calculated using Stefan Boltzmann law:

$$q = \varepsilon \, \sigma \, T_s^4 \tag{1.3}$$

where q is energy released per unit area (W/m^2), $\sigma = 5.67 \times 10^{-8} W/m^2/K^4$ is the Stefan Boltzmann constant, ε is the emissivity of the surface and T_s is the surface temperature (K).

When a small convex object is completely surrounded by another surface at a temperature T_{sur}, the net rate of radiation heat transfer from the object surface can be expressed as [13]:

$$q_{net} = \sigma \, A \, \varepsilon \, \left(T_s^4 - T_{sur}^4\right) \tag{1.4}$$

where A is the object area, ε is the object surface emissivity and T_s its temperature. In the case of I2MPECT project aiming to conceive a three-phase $45kW$ DC/AC converter, the target efficiency is 99%. The dissipated heat is equal to $450W$ for the 12×3 MOSFETs, which is equivalent to $12.5W$ per MOSFET. If we consider a surrounding medium at a temperature T_{sur} ranging from $40\ °C$ to $200\ °C$ and a chip with the upper surface dimensions $4\ mm$ by $6\ mm$, then we can see on Figure 1.8 that the amount of exchanged radiation heat does not exceed $0.1\ W$ which is much lower than the dissipated power. Therefore, the radiation heat transfer mode can be omitted for the I2MPECT power module.

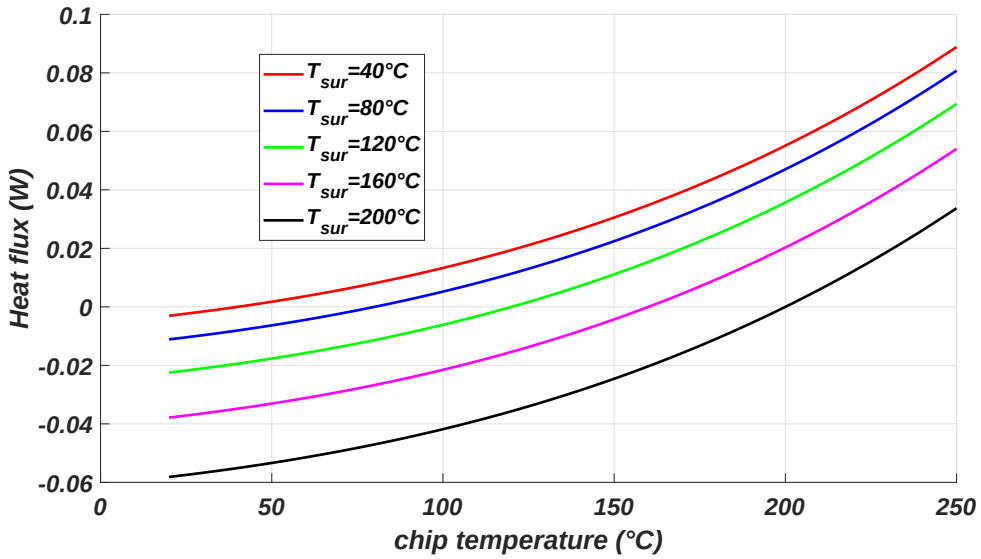


Figure 1.8 – Radiation heat transfer energy

1.2.2 Heat transfer in a typical power electronic module

The typical cooling solution for electronic boards (chips and DBC) consists of a heat spreader, a thermal interface material and a heat sink. This solution is presented on Figure 1.9. The two dominant heat transfer modes in this kind of package are heat conduction and convection. Heat conduction occurs when heat crosses the solid parts of the power module and flows toward the cold source. The dissipated energy is then removed by convection when a cooling fluid goes through heat fins and takes off all the heat. The heat path from the chip to the heat sink is presented on Figure 1.9.

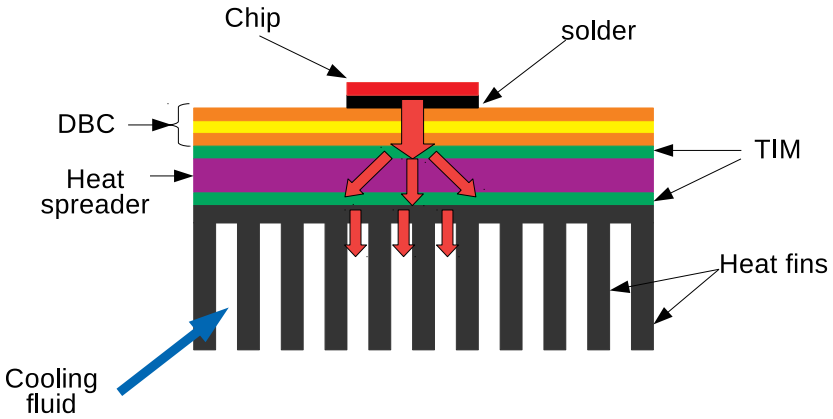


Figure 1.9 – Heat path in an electronic packaging

When heat crosses interfaces between elements of the power module, the heat will encounter a difficulty to move through the interface which will require a high temperature difference. This barrier is known as contact thermal resistance [15] and is reduced using thermal interface material as it will be depicted in section 1.3.2.

As heat moves from the MOSFET toward the cooling system, hot spots can apprear at the DBC surface. When putting the heat sink directly in contact to the DBC surface, fins which are not in front of these hot spots will evacuate less heat than the ones which are in front of hot spots. In addition putting thermal interface material on these hot spots will hasten its aging and thus will increase the contact thermal resistance [16]. In such situation, it is wise to homogenize the temperature before removing the heat by heat sink. Many technologies of heat spreading exist: solid heat spreader (section 1.3.6) or two-phase heat spreader (section 1.3.4).

1.3 Cooling solutions

In this section, we describe the most popular cooling techniques encountered in the literature.

1.3.1 Heat sink

A heat sink is a device used to remove the heat from electronic components by convection in order to control their temperature. As the heat sink contains many fins, the heat exchange area between the fluid and the solid is increased resulting in higher heat exchange rate. The shape of the heat fins and their arrangement are key factors in the performance of the heat sink. Many heat sink geometries exist (Figure 1.10).

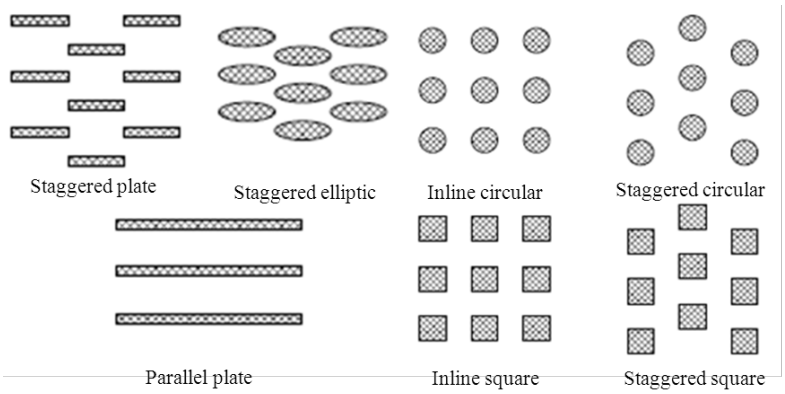


Figure 1.10 – Various geometries of heat sink [17]

Heat transfer performances of these geometries are studied in [17] and reveals that the best heat sink type is the staggered circular (Figure 1.11). It can be seen from Figure 1.11 that to reach the same heat transfer coefficient, staggered circular fins produce less pressure drop than the other geometries.

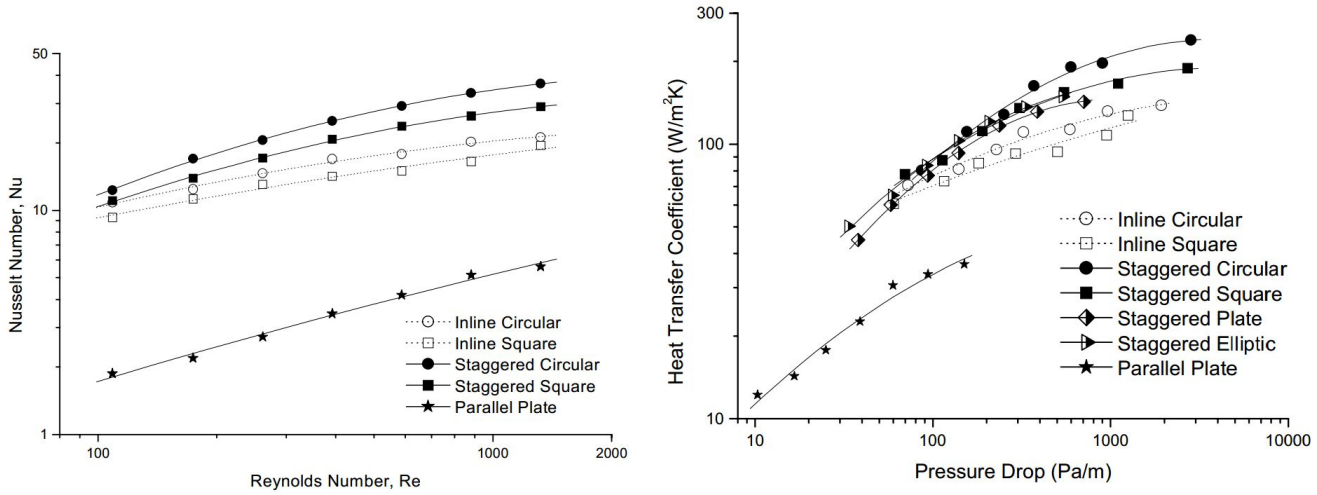


Figure 1.11 – Nusselt number versus Reynolds number for various fin geometry [17] (left), and the variation of the heat transfer coefficient versus pressure drop for various fin geometry [17] (right).

When designing a heat sink, we focus on minimizing both thermal resistance and pressure drop of the cooling fluid. Many optimization procedures have been used to determine optimal design parameters of the heat sink. For instance S.Ndao et al [18] used a genetic optimization algorithm and Pareto ranking method to determine optimal design parameters for various fin shape heat sink.

Other consideration like weight and compactness can be implemented during the optimization procedure [19,20]. Weight optimization of the heat sink is crucial when dealing with applications like MEA or electrical vehicles in which the overall system weight should be reduced in order to minimize the vehicle fuel consumption.

The minimization of the generated entropy can also be an interesting way for determining optimal design parameters of the pin fin heat sinks [21,22].

1.3.2 Thermal Interface Material (TIM) and contact thermal resistance

An electronic package is an assembly of many layers performing each one a relevant task. When assembling elements of electronic package directly with mechanical force (screws), micro gaps appear at the interface between the two solids (see Figure 1.12).

These gaps result in a contact thermal resistance that will stand against the energy flow. Thus a large temperature difference is required at the two solids interface in order to remove the heat from the chip to the cooling fluid (see Figure 1.12).

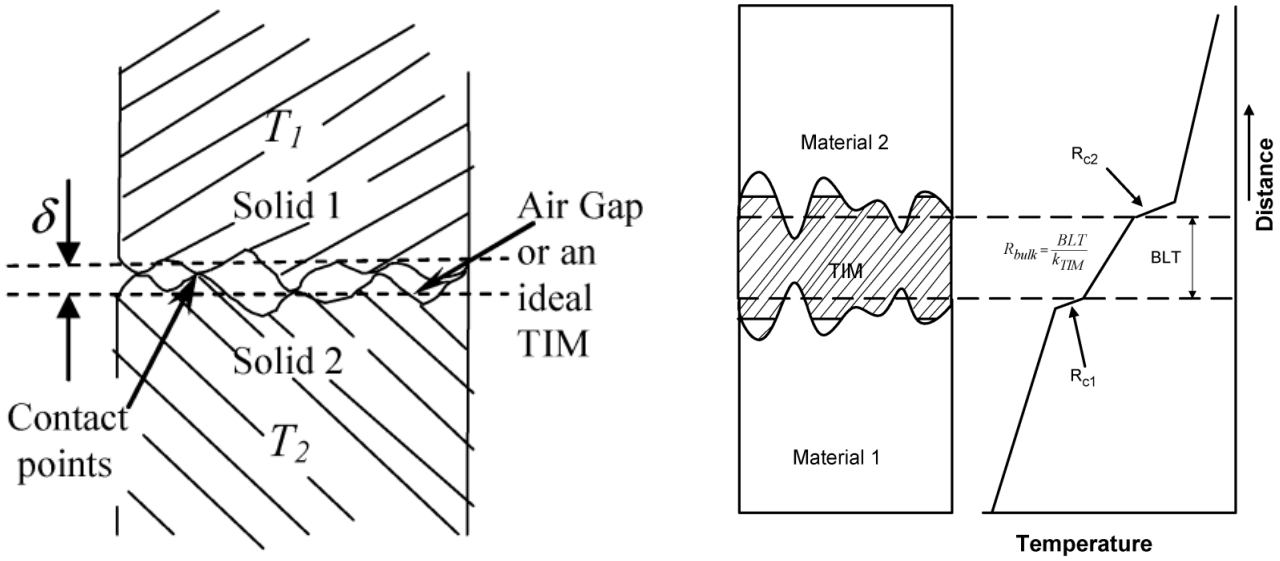


Figure 1.12 – Air gap at the assembly interface [23] (left), and the temperature gap across the contact thermal resistance (right)

Many models for contact thermal resistance calculation exist [24]. It is shown in [24] that the contact thermal resistance depends on the applied pressure and the roughness at the interface between the two solids. Contact thermal resistance decreases when the clamping pressure increases and the roughness decreases.

To bypass this problem we use thermal interface material (TIM) to fill the gap between the two solids and expel the interstitial air. In order to make this solution efficient, the thermal conductivity of the gap filler (TIM) should be greater than the air thermal conductivity. A

formula to calculate the resulting contact thermal resistance after adding the TIM is presented in [24]. This formula is based on the TIM thermal conductivity, the assembling pressure, the microhardness of the softer material and the roughness at the two solids joining interface.

Many types of TIM exist. The most popular ones are thermal grease, phase change material (PCM), thermally conductive elestomer and carbon based TIM.
Thermal grease consists of a polymer base and ceramic or metallic fillers. The base polymer material that is often used is silicon for its good thermal and mechanical properties. The main advantage of thermal grease is its lower contact thermal resistance compared to the other TIM solutions. The main drawbacks of thermal grease is the fact that it can be pumped out of the joining area and consequently it will mess its surrounding. The range of thermal grease contact thermal resistance is between $0.2\ K/cm^2/W$ and $1\ K/cm^2/W$ [25].

The phase change material as its name indicates is a material that melts at the operating temperature (in the $50 - 80\ °C$ range for thermoplastic adhesives [25]). When reaching the melting temperature the PCM will flow and fill all the air gaps which improves the heat flow at the interface between solids. The PCM technology offers many advantages like no pump out issue and higher stability. PCMs are also considered for their capability of reducing the thermal stress defined as the time derivative of the junction temperature, which will enhance the reliability of semi conductor devices. Drawbacks of PCM are the lower thermal conductivity compared to thermal grease and the required initial clamping pressure that will lead to high mechanical stress. To enhance the thermal conductivity of PCM some studies suggest to add carbon nanotubes (CNTs) to the PCM [26]. The range of PCM contact thermal resistance is between $0.3\ K/cm^2/W$ and $0.7\ K/cm^2/W$ [25].

When exposed to temperature cycling the TIM can dry out which will form air gaps inside it. This fact will increase the contact thermal resistance. This phenomenon was investigated in [16] on several TIM materials (Polyimide PCM, graphite sheet, elastomer film) and the obtained results revealed that only for the Polyimde PCM a decrease of the thermal conductivity of $30\ \%$ is observed when undergoing 1500 temperature cycles $(-50\ °C/150\ °C)$.

1.3.3 Heat pipe and Thermosyphon

The heat pipe and thermosyphon are similar two-phase systems used to transfer the heat from a hot source called evaporator to a cold source called condenser. These devices can also be used to spread out the hot spots and homogenize the temperature [27]. The fluid inside these systems is set at saturation conditions. The main advantages of these cooling system is that they don't require pumping power and don't include mechanical moving parts. In practice, the device dissipating heat (electronic components or the power module) is placed on the evaporator. The condenser is joined to a heat sink.

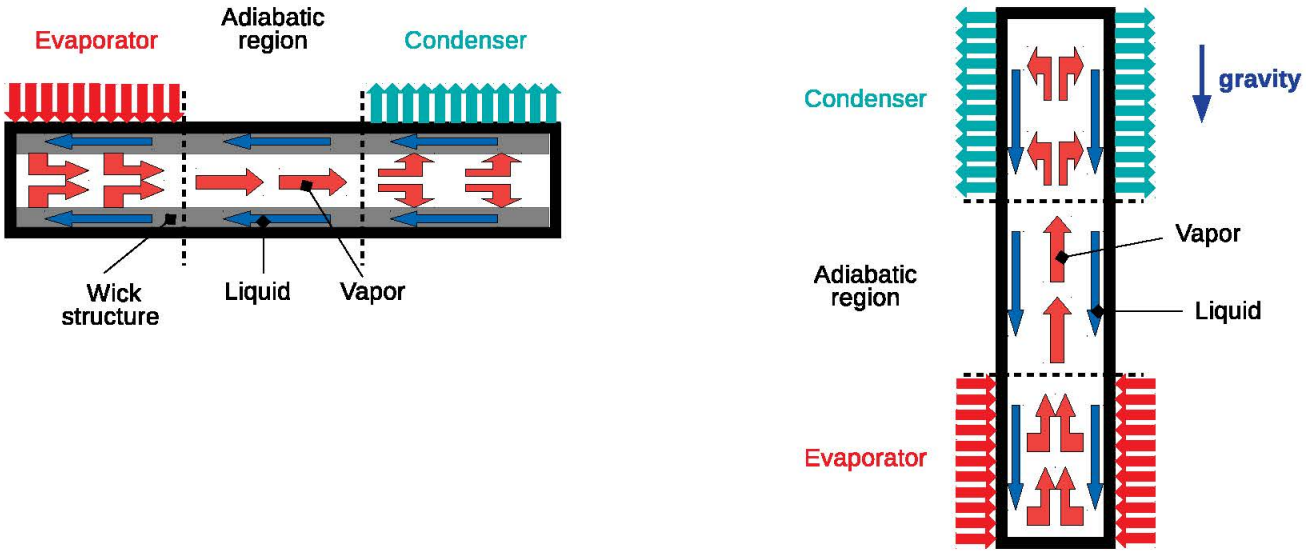


Figure 1.13 – Heat pipe (left), and thermosyphon (right)

The difference between the heat pipe and the thermosyphon lies in the mechanism that move the condensed liquid from the condenser to the evaporator (see Figure 1.13). In the heat pipe this force is created by capillary mechanism of the wick structure. Capillary forces depend on the operating condition and the wick geometrical and physical properties. The wick structure can be porous material or grooves. The capillary force can be calculated by the Young Laplace equation:

$$\Delta P_{cap} = \frac{2\sigma}{r}$$

where r is minimum possible radius of curvature at the liquid vapor interface in the porous structre and σ is the fluid surface tension. The widely used working fluid are water, acetone, methanol and ethanol. The chemical compatibility between the working fluid and the wick structure material should be checked in order to avoid degradation of the two phase cooling (corrosion, generation of non condensable gases inside the two phase system) [28]. In the thermosyphon, the condensed liquid returns back to the condensor thanks to gravity. In each case, the created force should balance the pressure drop when the fluid goes from the condenser to the evaporator. In some applications, like cooling of a motor core, contrifugal force (see Figure 1.14) is used to return back the fluid to the condensor [29, 30]. As shown in Figure 1.14 the difference in the internal radius between the evaporator and the condenser makes the fluid slip and return back to the condenser.

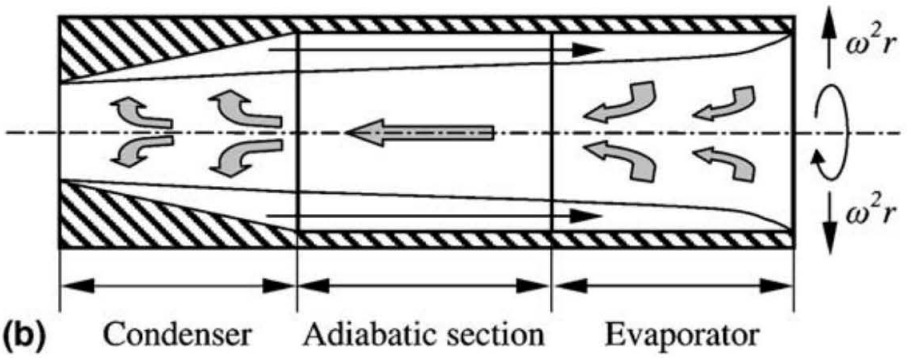


Figure 1.14 – Rotating heat pipe [29]

Operating limits of the two-phase cooling system should be studied to make sure that electronic components are continuously cooled. When one of these limits are met, the heat flow from the evaporator to the condensor is blocked and the high temperature leads to failure of the system to cool. These limits [31–33] are:

1. **Viscous limit**: This limitation corresponds to the case where the viscous forces of the working fluid are greater than the vapor pressure gradient created by the heat load. This unbalance in forces stands against the fluid flow and consequently against energy transfer between the evaporator and the condensor.

2. **Sonic limit**: When the shape of the liquid looks like a converging diverging nozzle, the vapor velocity can reach the speed of sound, which will produce a shockwave inside the pipe. This shockwave will block the vapor flow between the condensor and the evaporator and can damage the internal structure of the pipe.

3. **Entrainment limit**: It can be seen from Figure 1.13 that liquid and vapor flows in opposite directions. As a result, shear forces take place at the interface between the two phases and can inhibit the return of liquid to the evaporator.

4. **Capillary limit**: This limit is related to the liquid flow inside the wick structure. When the pressure drop inside the pipe exceeds the pressure created by the capillary structure the flow of the fluid stops. In such situations, the wick structure dries out and the heat pipe operation becomes unstable.

5. **Boiling limit**: When heat is submitted to the evaporator, a boiling phenomenon occurs and bubbles are created in the wick structure of the evaporator. These bubbles can growth in a way that make the return of the liquid to evaporator impossible. Thus, the wick can dry out and temperature of the evaporator will increase sharply.

1.3.4 Loop Heat Pipe (LHP)

In contrast to the heat pipe and thermosyphon, the evaporator and condenser in the loop heat pipe are separated by pipes. The aim of this conception is to transfer the heat over a greater distance which is needful when the hot and cold source are far apart. A schematic representation of a LHP is shown in Figure 1.15.

A LHP consists of four components: the evaporator, the condenser, the liquid line and the vapor line.
The evaporator is the central element of a LHP. It's where heat is dissipated and where capillary forces are generated to pump the fluid inside the LHP. The evaporator consists of a porous wick, vapor grooves and a reservoir that can be integrated inside the evaporator (traditional LHP) or in the liquid line (Capillary Pumped Loop CPL) [34].
In the condenser, heat is removed using a heat exchanger. The vapor is cooled until it becomes entirely liquid.

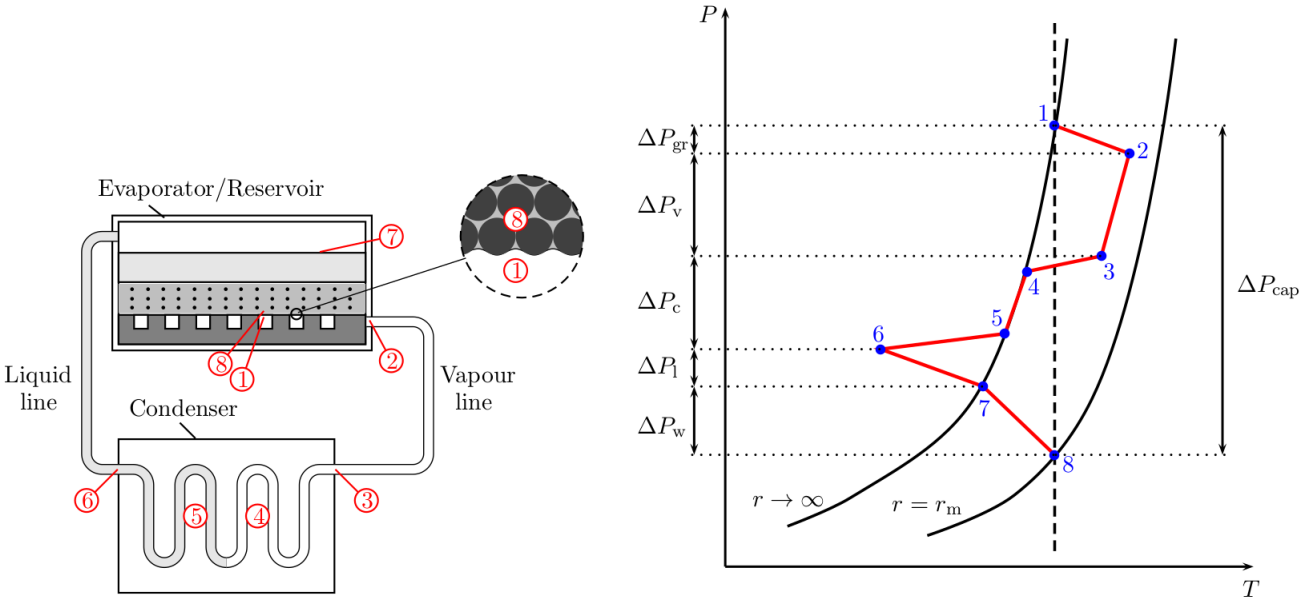


Figure 1.15 – Scheme of a loop heat pipe (left), and diagram of an LHP operating cycle [35] (right).

The thermodydnamic behavior of the LHP is described by the (P, T) diagram presented on Figure 1.15. It can be seen that when the fluid crosses the LHP, different kind of pressure drops occurs which should be balanced by the capillary force created by the wick structure (ΔP_{cap}).

The thermal behavior of a LHP depends on operating conditions in particular the heat load at the evaporator. A typical trend of the LHP heat transfer coefficient is presented on Figure 1.16.

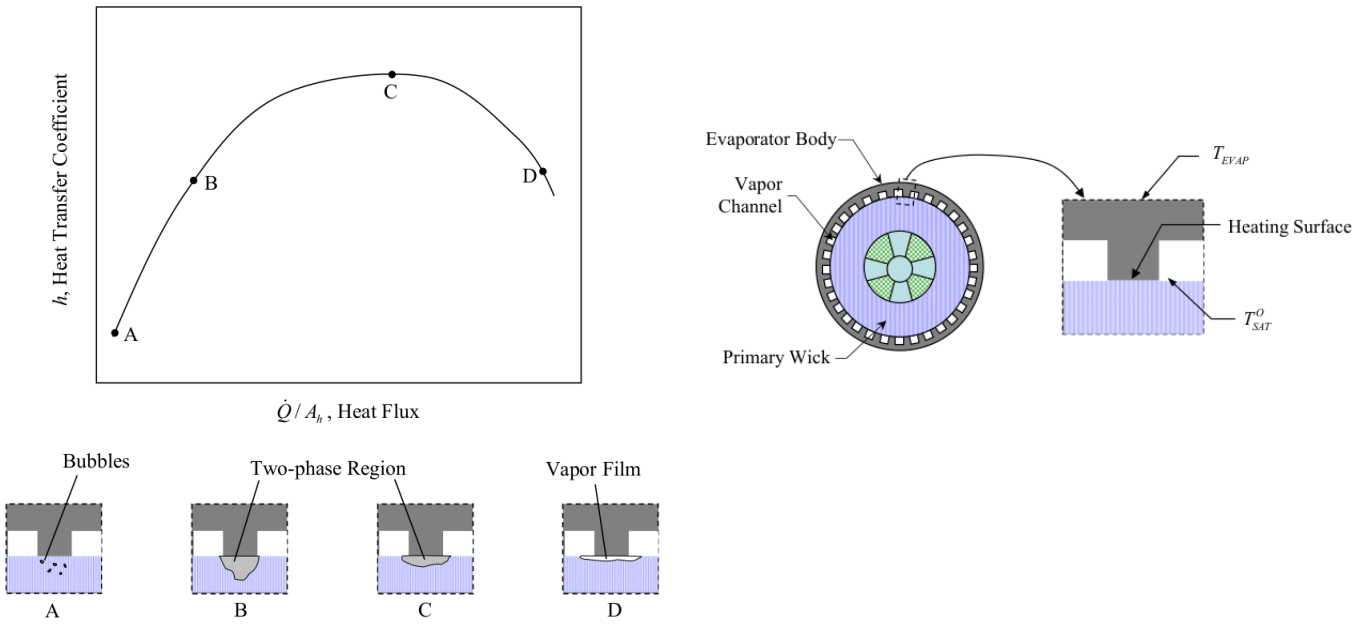


Figure 1.16 – Typical heat transfer coefficient of an LHP and a drawing of the two phase region [32,36]

The evolution of the thermal performance of the LHP can be explained by the type of two phase flow appearing at the interface between the wick structure and the vapor grooves [32,36]. The interdependence between the LHP performance and the type of the two phase flow can be seen on Figure 1.16.

Other factors affect the performance of a LHP like the inclination with respect to gravity [37,38] and the acceleration [38–40].

An analytical thermal model of LHP is developed in [31]. This model allows prediction of the reservoir temperature as a function of geometrical parameters and fluid thermophysical properties. The drawback of this model is that it requires the identification of the evaporator thermal resistance which is a priori unknown. Once the evaporator thermal resistance is determined, its temperature, which is the quantity of interest in applications, can be deducted .

1.3.5 Jet Impingement

Jet impingement is used when dealing with electronic components dissipating high amounts of heat. Jet impingement consists in ejecting at high speed a cooling fluid onto a hot surface (electronic components) like shown in Figure 1.17.

There exist two configurations of jet impingement:

1. The unconfined submerged jet .
2. The confined submerged jet in which the flow is bounded by two plates as shown in Figure 1.17.

It is known that the second family of jet impingement offers better thermal performance.

When fluid hits a high temperature surface, evaporation or boiling of the coolant can occur. In such case the heat transfer rate is improved [41].

The effectiveness of this cooling solution depends on many factors such as the fluid nature, the nozzle to target spacing distance (L_y on Figure 1.17), the jet diameter (D on Figure 1.17), the jet velocity and the fluid properties.

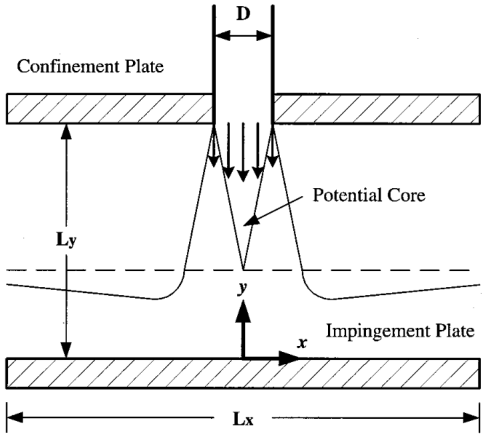


Figure 1.17 – Confined submerged jet [42]

The main drawback of this cooling solution is the concentrated heat transfer at the stagnation point (see Figure 1.18). This will restrict the cooling to a narrow region of the system (near the

stagnation point). To bypass this limit, some researchers have studied the use of spray cooling [43,44].

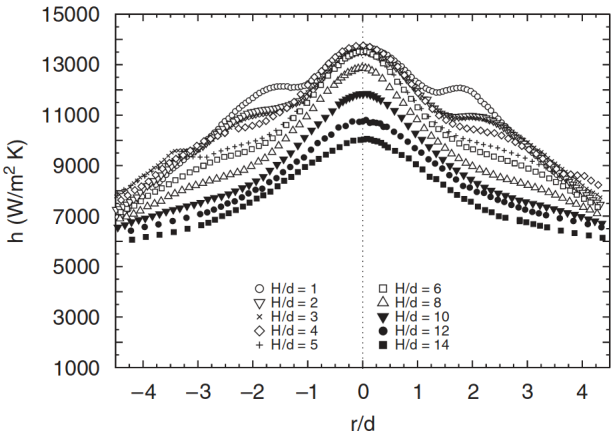


Figure 1.18 – Variation in local heat transfer coefficient distribution with jet to target spacing (H/d) for $Re = 13000$ and $d = 1.59\ mm$. [45]

It can be observed from Figure 1.18 that when increasing the nozzle to target distance the heat transfer coefficient increases. This distance can be adjusted with respect to the dissipated power and the maximum allowed temperature.

1.3.6 Solid heat spreader

As outlined in section 1.2.2 it is important to spread heat before evacuating it by convection. The use of a heat spreader makes all fins of the heat sink functional and consequently increases the efficiency of the cooling system. As mentioned in section 1.3.3 two-phase cooling devices have operating limits that can lead to electronic failure. Controlling operating limits is not easy to perform. That is why in some situation it is wise to use solid heat spreaders. It is possible to spread heat using a simple base plate or thermally conductive sheet.

Pyrolytic graphite sheet (PGS) is an anisotropic material that has a high in-plane thermal conductivity and low through-thickness thermal conductivity. This property allows it to spread heat and minimize the hot spots. PGS has been used to manage the heat inside a proton exchange membrane fuel cell [46]. Such technique lowers the temperature inside the device [46].

Graphite spreader exhibits good thermal performance when dealing with handheld device [47] by reducing the temperature magnitude and gradient (see Figure 1.19).

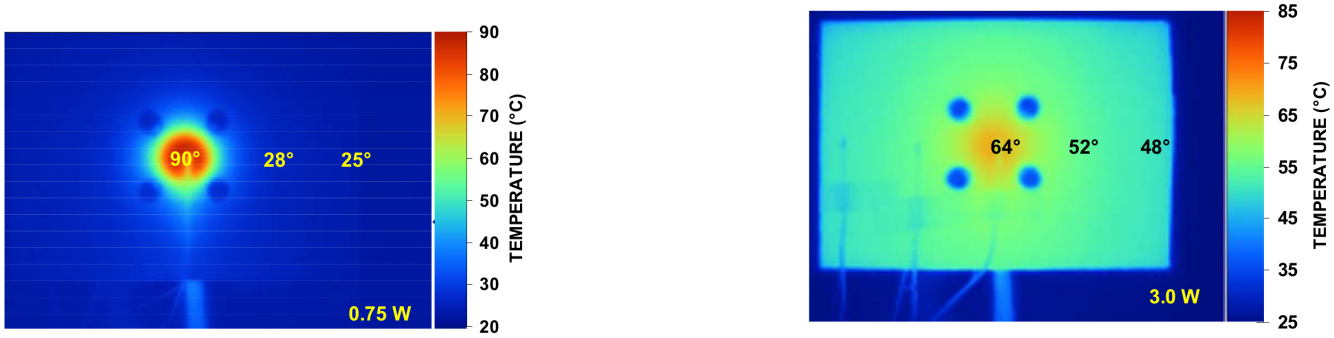


Figure 1.19 – Temperature field without (Left) and with (Right) graphite heat spreader [47]

Yin Xiong [47] compared three spreading solutions (Graphite spreader, copper and aluminium base plate) and showed that using graphite spreader is better than the two other spreading methods.

1.3.7 Thermoelectric cooling

Thermoelectric cooling is based on the Peltier thermo electric effect. The Peltier effect is the conversion of a temperature difference to electric current and vice versa. This effect is used to enforce a temperature difference by feeding the thermoelectric refrigerator with electricity (see Figure 1.20). A thermoelectric refrigerator consists of two plates: the cold plate and the hot plate. In most cases, the cold side of thermoelectric device is joined to the electronic device to be cooled whereas the hot side is connected to a convection cooled heat sink (see Figure 1.20).

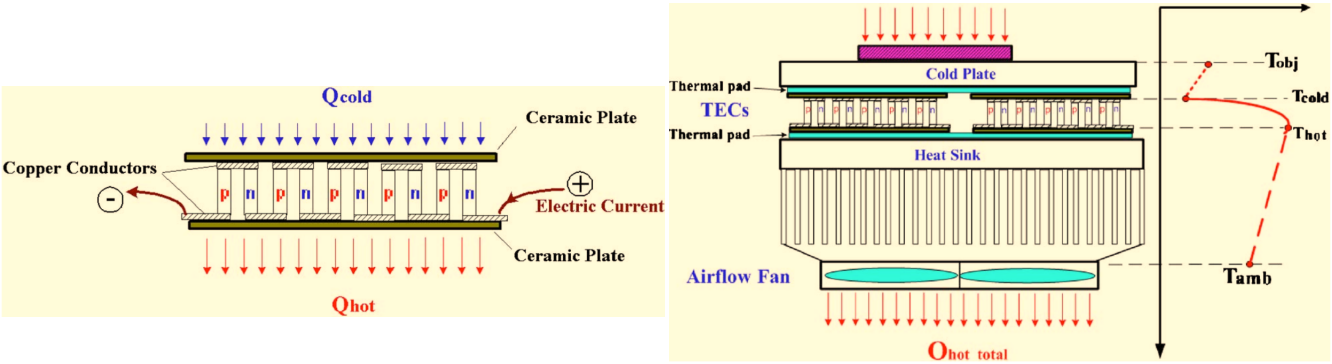


Figure 1.20 – Scheme of thermoelectric network (left), and typical scheme of the thermoelectric cooling device [48] (right).

Thermoelectric coolers have many advantages such as: low maintenance effort, absence of mechanical moving parts and refrigerant and its capability to support shocks and high levels of vibrations [49,50].

The disadvantages of this technology are the low coefficient of performance (less than 1 when temperature difference exceeds $20°C$) and its need electric energy to perform its task [49,50].

1.4 The chosen cooling solution and the studied system

1.4.1 The chosen cooling solution

Many considerations have been taken into account in order to choose the suitable cooling technique. These considerations depend on the system that we have to cool, the application requirements as well as the surrounding conditions.

The system to be cooled is an electronic DC/AC power converter that will be used for aerospace application. In this kind of application, the reliability of the system under various use conditions is a central requirement.

Moreover, when the external temperature is negative, two-phase cooling systems can be inefficient during the starting operating period because the internal working fluid can be in solid state. This will claim a period of time during which the working fluid will melt and be able to evacuate heat from electronic components. The problem is that melting period can be long and thus electronic component temperature can exceed the maximum allowed temperature. This phenomenon is known as blowby freezing and was investigated experimentally in [51, 52] for heat pipes application. Therefore the two-phase spreading technique was not selected for this application. Otherwise, research on the efficiency the two-phase heat spreader have been maintained within the I2mpect project and promising results have been obtained.

Another constraint to be considered is the available space for the cooling system and its location inside the cabinet with respect to the cooling air inlet duct. A picture of the power module with the cooling system is depicted in Figure 1.21.

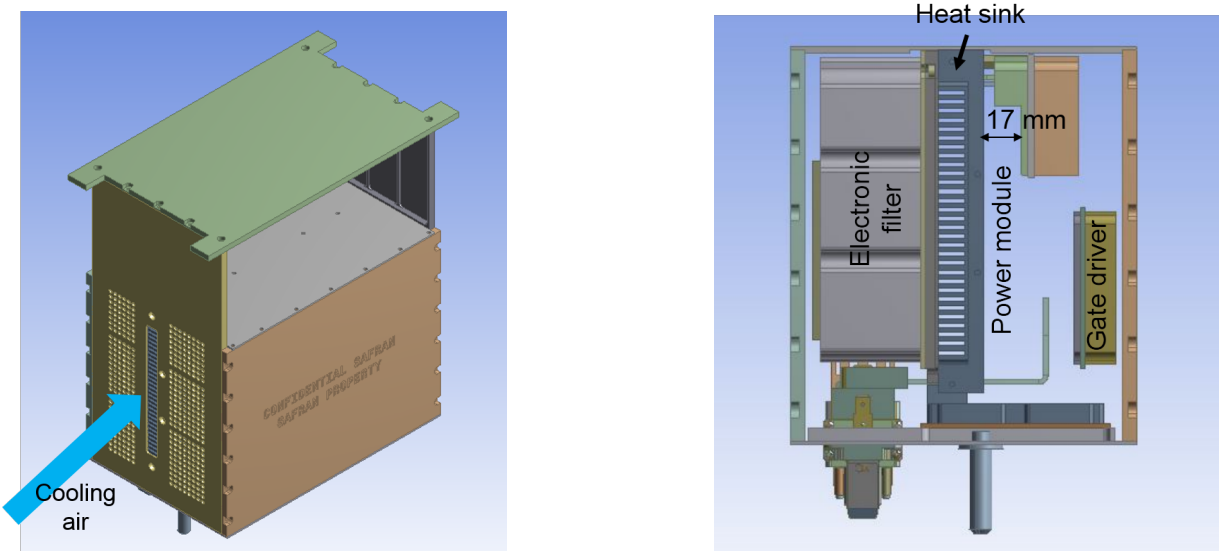


Figure 1.21 – The cabinet enclosure and the air inlet duct (left), and the allowed space for the power module (right).

All these considerations lead to choosing a conventional cooling system consisting of an aluminium base plate and a straight fins heat sink. The choice of straight fins can be explained by their good trade-off between low pressure drop and good heat transfer rate.

Efficiency of the chosen cooling system for industrial operating conditions is verified by numerical simulation and is in good agreement with specifications.

1.4.2 The studied system

The study will focus on the assembly power module and the cooling system without considering the influence of the cabinet and the surrounding components. The three-leg power converter with the heat sink is presented in Figure 1.22.

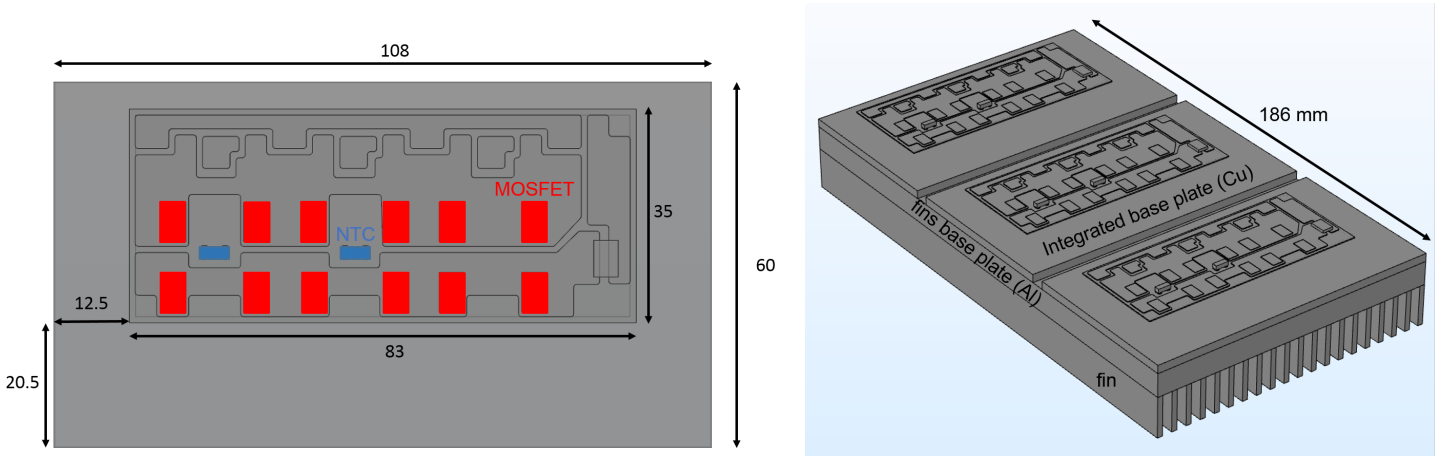


Figure 1.22 – The upper view and dimensions of the one leg converter (left) and the three legs converter with the cooling system (right).

In this configuration, the coupled convection and conduction problem should be solved in order to find the temperature distribution within the system. Such procedure provides realistic results but makes the manipulation of the finite element matrices very complex.

In fact, the commercial finite element software $COMSOL^{®}$ has been used to solve the heat transfer problem in the power module. By using this software, the finite element matrices have been recovered. In the case of conductive heat transfer with natural boundary conditions (convection coefficient set on a part of the power module external surface), applying time integration method on the recovered matrices yields results identical to $COMSOL^{®}$.

In contrast, when treating conjugate heat transfer, we were not able to reproduce the commercial software results using the recovered matrices.

In order to find a relevant value of boundary condition, the cooling system is modeled by an equivalent heat coefficient h_{eq} as shown in Figure 1.23. This approach is used to construct the direct reduced order model presented in chapter 4.

The inverse reduced order model (IROM) that will be presented in chapter 5 does not require to know finite element matrices. The only required information to construct the IROM is the temperature field computed for a given operating conditions. Therefore, the IROM will be constructed for the case of conjugate heat transfer.

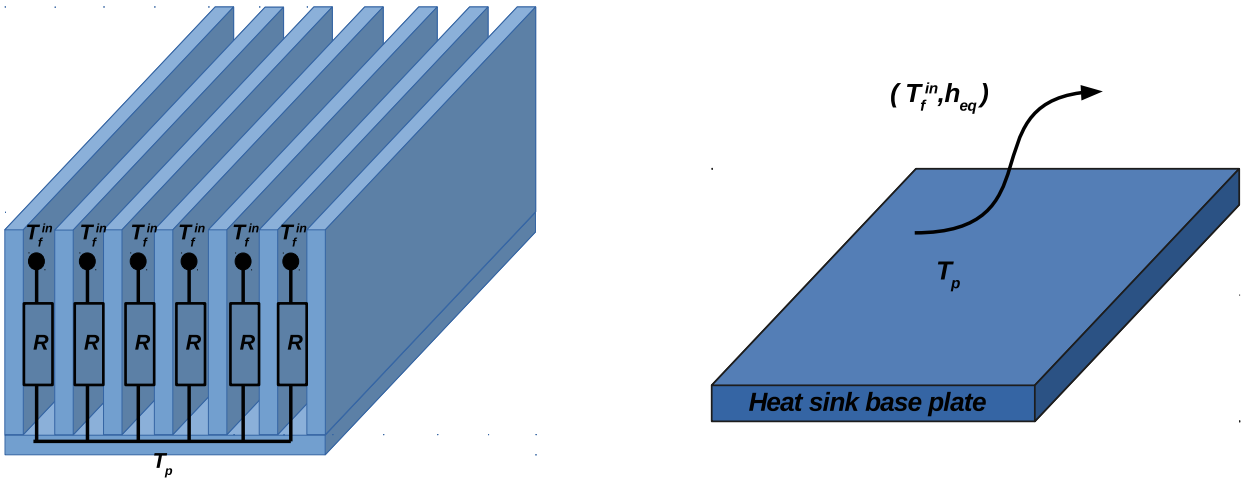


Figure 1.23 – Heat sink thermal network (left), and the Equivalent convective boundary condition (right).

The equivalent heat coefficient can be obtained by equation (1.5) :

$$h_{eq} = \frac{(N-1)\,\dot{m}\,C_p\,\left[1 - exp(\frac{-A_{eff}\;h_m}{\dot{m}C_p})\right]}{S_{eq}} \tag{1.5}$$

where N is the number of heat fins, $\dot{m}$ is the mass flow rate entering a single channel, C_p is the heat capacitance of the air, h_m is the mean convection coefficient (see Appendix A), A_{eff} is the effective heat exchange area of a single fin (see Appendix A) and S_{eq} is the heat sink base plate surface. Calculation details of h_{eq} are given in Appendix A.

1.5 Conclusion

In this chapter, a description of the studied PEM structure and operation is given. Power losses occuring in MOSFETs are illustrated and its dependance with respect to switching frequency and electrical signals is outlined. The strong relationship between failure in electronic devices and temperature is depicted. A relation linking failure rate and junction temperature was presented in this chapter.

Therefore, it is needful to manage the heat inside the PEM. A description of heat transfer modes is presented as well as a description of energy path inside a typical PEM. Then different cooling solutions are described. Advantages and drawbacks of each cooling solution are highlighted.

Industrial specifications are described briefly in order to show the relevance of the chosen cooling solution. The chosen cooling solution consists of a solid heat spreader (base plate) attached to a straight-fin heat sink.

Finally, the thermal resistance of the chosen heat sink is calculated as a function of geometric properties, physical properties of the fluid and the fin, and operating conditions (fluid inlet velocity and temperature). An equivalent heat transfer coefficient is derived from this thermal resistance in order to set a relevant boundary condition to the PEM. The usefulness of this coefficient will be noticed in the next chapters when constructing the thermal finite element model of the PEM.

Bibliography

[1] Gangyao Wang, Fei Wang, Gari Magai, Yang Lei, Alex Huang, and Mrinal Das. Performance comparison of 1200v 100a sic mosfet and 1200v 100a silicon igbt. In *Energy Conversion Congress and Exposition (ECCE), 2013 IEEE*, pages 3230–3234. IEEE, 2013.

[2] Ljubisa D Stevanovic, Kevin S Matocha, Peter A Losee, John S Glaser, Jeffrey J Nasadoski, and Stephen D Arthur. Recent advances in silicon carbide mosfet power devices. In *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, pages 401–407. IEEE, 2010.

[3] Allen R Hefner. A dynamic electro-thermal model for the igbt. In *Industry Applications Society Annual Meeting, 1992., Conference Record of the 1992 IEEE*, pages 1094–1104. IEEE, 1992.

[4] Jin Ye, Kai Yang, Haizhong Ye, and Ali Emadi. A fast electro-thermal model of traction inverters for electrified vehicles. *IEEE Transactions on Power Electronics*, 32(5):3920–3934, 2017.

[5] WANG HUANHUAN. *Dynamic Electro-Thermal Model of Power Electronic Building Block in Micro Grid: Modeling, Analysis and Simulation*. PhD thesis, 2012.

[6] Dusan Graovac, Marco Purschel, and Andreas Kiep. Mosfet power losses calculation using the data-sheet parameters. *Infineon application note*, 1, 2006.

[7] Dustin A Murdock, Jose E Ramos Torres, Jeffrey J Connors, and Robert D Lorenz. Active thermal control of power electronic modules. *IEEE Transactions on Industry Applications*, 42(2):552–558, 2006.

[8] Michael Pecht. *Handbook of electronic package design*, volume 76. CRC Press, 1991.

[9] Jonas Arwidson. *Thermal Fatigue Life Prediction of Solder Joints in Avionics by Surrogate Modeling: A Contribution to Physics of Failure in Reliability Prediction*. PhD thesis, LiU-Tryck, Sweden, 2013.

[10] Military Handbook. Electronic reliability design handbook. Technical report, MIL-HDBK-338, DoD, 1988.

[11] Pradeep Lall, Michael Pecht, and Edward B Hakim. *Influence of temperature on microelectronics and system reliability: A physics of failure approach*. CRC press, 1997.

[12] Douglas A Pietila, Mostafa Rassaian, and Ken Brice-Heames. Design characterization of microwave antenna bga interconnect structure using test-validated physics-of-failure methods. In *Reliability Physics Symposium Proceedings, 1999. 37th Annual. 1999 IEEE International*, pages 347–355. IEEE, 1999.

[13] Theodore L Bergman, Frank P Incropera, David P DeWitt, and Adrienne S Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[14] Yunus Cengel. *Heat and mass transfer: fundamentals and applications*. McGraw-Hill Higher Education, 2014.

[15] MM Yovanovich, JR Culham, and P Teertstra. Calculating interface resistance. *Electronics Cooling*, 3(2):24–29, 1997.

[16] Jean-Pierre Ousten and Zoubir Khatir. Study of thermal interfaces aging for power electronics applications. In *Power Electronics and Applications (EPE 2011), Proceedings of the 2011-14th European Conference on*, pages 1–10. IEEE, 2011.

[17] Denpong Soodphakdee, Masud Behnia, and David Watabe Copeland. A comparison of fin geometries for heatsinks in laminar forced convection—part i: Round, elliptical, and plate fins in staggered and in-line configurations. *Int. J. Microcircuits Electron. Packag*, 24(1):68–76, 2001.

[18] Sidy Ndao, Yoav Peles, and Michael K Jensen. Multi-objective thermal design optimization and comparative analysis of electronics cooling technologies. *International Journal of Heat and Mass Transfer*, 52(19-20):4317–4326, 2009.

[19] Christoph Gammeter, Florian Krismer, and Johann W Kolar. Weight optimization of a cooling system composed of fan and extruded-fin heat sink. *IEEE Transactions on Industry Applications*, 51(1):509–520, 2015.

[20] Puqi Ning, Guangyin Lei, Fred Wang, and Khai DT Ngo. Selection of heatsink and fan for high-temperature power modules under weight constraint. In *Applied Power Electronics Conference and Exposition, 2008. APEC 2008. Twenty-Third Annual IEEE*, pages 192–198. IEEE, 2008.

[21] Waqar A Khan, J Richard Culham, and M Michael Yovanovich. Optimization of pin-fin heat sinks using entropy generation minimization. *IEEE Transactions on Components and Packaging Technologies*, 28(2):247–254, 2005.

[22] Ken Ogiso. Assessment of overall cooling performance in thermal design of electronics based on thermodynamics. *Journal of heat transfer*, 123(5):999–1005, 2001.

[23] Ravi Prasher. Thermal interface materials: historical perspective, status, and future directions. *Proceedings of the IEEE*, 94(8):1571–1586, 2006.

[24] MM Yovanovich and EE Marotta. Thermal spreading and contact resistances. *Heat Transfer Handbook*, 1:261–394, 2003.

[25] Daniel Blazej. Thermal interface materials. *Electronics Cooling*, 9:14–21, 2003.

[26] Emmanuel Ollier, U Soupremanien, V Remondière, J Dijon, H Le Poche, AL Seiler, F Lefevre, S Lips, C Kinkelin, Nathalie Rolland, et al. Thermal management of electronic devices by composite materials integrated in silicon. *Microelectronic Engineering*, 127:28–33, 2014.

[27] Nicolas Blet, Stéphane Lips, and Valérie Sartre. Heats pipes for temperature homogenization: A literature review. *Applied Thermal Engineering*, 118:490–509, 2017.

[28] D Mishkinis, P Prado, R Sanz, A Torres, AS Merino, and T Tjiptahardja. Development of lhp for intermediate temperature range. In *15th International Heat Pipe Conference, Clemson, South Carolina (USA)*, 2010.

[29] F Song, D Ewing, and CY Ching. Experimental investigation on the heat transfer characteristics of axial rotating heat pipes. *International Journal of Heat and Mass Transfer*, 47(22):4721–4731, 2004.

[30] TC Jen, G Gutierrez, S Eapen, G Barber, H Zhao, PS Szuba, J Labataille, and J Manjunathaiah. Investigation of heat pipe cooling in drilling applications.: part i: preliminary numerical analysis and verification. *International Journal of Machine Tools and Manufacture*, 42(5):643–652, 2002.

[31] Stéphane Launay, Valérie Sartre, and Jocelyn Bonjour. Parametric analysis of loop heat pipe operation: a literature review. *International Journal of Thermal Sciences*, 46(7):621–636, 2007.

[32] Po-Ya Abel Chuang. An improved steady-state model of loop heat pipes based on experimental and theoretical analyses. 2003.

[33] MOHAMMAD OMAR HAMDAN. *Loop heat pipe (Lhp) modeling and development by utilizing coherent porous silicion (Cps) wicks*. PhD thesis, University of Cincinnati, 2003.

[34] Dan Butler, Jentung Ku, and Theodore Swanson. Loop heat pipes and capillary pumped loops-an applications perspective. In *AIP Conference Proceedings*, volume 608, pages 49–56. AIP, 2002.

[35] Benjamin Siedel. *Analysis of heat transfer and flow patterns in a loop heat pipe: Modelling by analytical and numerical approaches and experimental observations*. PhD thesis, INSA de Lyon, 2014.

[36] Quan Liao and TS Zhao. Evaporative heat transfer in a capillary structure heated by a grooved block. *Journal of Thermophysics and Heat Transfer*, 13(1):126–133, 1999.

[37] Xiang-you Lu, Tse-Chao Hua, Mei-jing Liu, and Yuan-xia Cheng. Thermal analysis of loop heat pipe used for high-power led. *Thermochimica Acta*, 493(1-2):25–29, 2009.

[38] Tarik Kaya and Jentung Ku. Thermal operational characteristics of a small-loop heat pipe. *Journal of Thermophysics and Heat Transfer*, 17(4):464–470, 2003.

[39] Jentung Ku, Laura Ottenstein, Tarik Kaya, Paul Rogers, and Craig Hoff. Testing of a loop heat pipe subjected to variable accelerating forces, part 2: Temperature stability. Technical report, SAE Technical Paper, 2000.

[40] Jentung Ku, Laura Ottenstein, Mark Kobel, Paul Rogers, and Tarik Kaya. Temperature oscillations in loop heat pipe operation. In *AIP Conference Proceedings*, volume 552, pages 255–262. AIP, 2001.

[41] Lu Qiu, Swapnil Dubey, Fook Hoong Choo, and Fei Duan. Recent developments of jet impingement nucleate boiling. *International Journal of Heat and Mass Transfer*, 89:42–58, 2015.

[42] Yongmann M Chung and Kai H Luo. Unsteady heat transfer analysis of an impinging jet. *Journal of heat transfer*, 124(6):1039–1048, 2002.

[43] Gangtao Liang and Issam Mudawar. Review of spray cooling–part 1: Single-phase and nucleate boiling regimes, and critical heat flux. *International Journal of Heat and Mass Transfer*, 115:1174–1205, 2017.

[44] J Yang, LC Chow, and MR Pais. Nucleate boiling heat transfer in spray cooling. *Journal of Heat Transfer*, 118(3):668–671, 1996.

[45] Suresh V Garimella and RA Rice. Confined and submerged liquid jet impingement heat transfer. *Journal of Heat Transfer*, 117(4):871–877, 1995.

[46] Chih-Yung Wen and Guo-Wei Huang. Application of a thermally conductive pyrolytic graphite sheet to thermal management of a pem fuel cell. *Journal of Power Sources*, 178(1):132–140, 2008.

[47] Yin Xiong, Martin Smalc, Julian Norley, John Chang, Helen Mayer, Prathib Skandakumaran, and Brad Reis. Thermal tests and analysis of thin graphite heat spreader for hot spot reduction in handheld devices. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, pages 583–590. IEEE, 2008.

[48] Boris Abramzon. Numerical optimization of the thermoelectric cooling devices. *Journal of Electronic Packaging*, 129(3):339–347, 2007.

[49] SB Riffat and Xiaoli Ma. Improving the coefficient of performance of thermoelectric cooling systems: a review. *International Journal of Energy Research*, 28(9):753–768, 2004.

[50] Dongliang Zhao and Gang Tan. A review of thermoelectric cooling: materials, modeling and applications. *Applied Thermal Engineering*, 66(1-2):15–24, 2014.

[51] JM Ochterbeck and GP Peterson. Investigation of the freezing blowby phenomenon in heat pipes. *Journal of thermophysics and heat transfer*, 9(2):314–321, 1995.

[52] Jean Michel Tournier and Mohamed S El-Genk. A vapor flow model for analysis of liquid-metal heat pipe startup from a frozen state. *International journal of heat and mass transfer*, 39(18):3767–3780, 1996.

Chapter 2

Finite element simulation of heat transfer in the power module

To study the thermal behavior of an electronic package, the temperature field should be computed. Governing equations of heat transfer in both fluid and solid media will be described. These equations do not have an analytical solution in the case of industrial systems such as the I2MPECT power module. Therefore numerical methods are needed to overcome this limit and calculate an approximation of the temperature. The finite element method is chosen to discretize in space the energy and Navier-Stokes equations. Finaly, the used finite element software will be presented as well as the reasons for this choice.

Sommaire

2.1 Governing equations of heat transfer in the power electronic module

In this section, a description of heat transfer in both fluid and solid media is given. The solid medium in the electronic package is the electronic board and the cooling system (base plate and heat fins). The fluid part is the cooling air crossing heat fins.

2.1.1 Heat transfer in the solid medium

The time and space variation of the temperature in a solid medium is descibed by equation (2.1), obtained from an energy balance [1]:

$$\rho_s.C_{p,s}.\frac{\partial T_s(\mathbf{x},t)}{\partial t} - \nabla.(k_s.\nabla T_s(\mathbf{x},t)) = Q(\mathbf{x},t) \tag{2.1}$$

where $T_s(\mathbf{x},t)$ is the temperature in the solid, ρ_s is the solid density, $C_{p,s}$ is the solid specific heat, k_s is the solid thermal conductivity and $Q(\mathbf{x},t)$ is the volumetric heat source. $(\mathbf{x},t)$ stands for position vector and time. $(\mathbf{x},t) \in \Omega_s \times]0,t_e[$ where Ω_s is the solid domain and $]0,t_e[$ is the time interval for studying the physical phenomenon.

All thermo-physical properties are assumed temperature-independant. Althought, they can be piecewise constant in space, as the electronic package contains many materials.

In order to solve equation (2.1), it is necessary to define boundary conditions (BCs). BCs can be either a known heat flux density or a known temperature. In our case, a convective boundary condition is considered and can be written as:

$$-k_s \nabla T_s.\mathbf{n} = h(T_s(\mathbf{x},t) - T_{ext}) \tag{2.2}$$

where T_{ext} is the external environment temperature, h is a heat exchange coefficient and $\mathbf{n}$ is the outward pointing unit normal vector of the boundary $\partial\Omega_s$. $\mathbf{x} \in \partial\Omega_s$ where $\partial\Omega_s$ is the boundary of the solid domain Ω_s submitted to heat exchange. This boundary condition can change over time, for instance when the fluid temperature or the heat transfer coefficient evolve with time. In the case of power electronic module cooled by heat sink, the heat transfer coefficient h is calculated in Appendix A.

As equation (2.1) depends on time, an initial condition is required and is given as follows:

$$T_s(\mathbf{x}, t=0) = T_{s0}(\mathbf{x}) \tag{2.3}$$

where $T_{s0}(\mathbf{x})$ in the initial temperature field in the solid domain Ω_s.

2.1.2 Heat transfer in the fluid medium

In a fluid medium, the temperature is coupled to the fluid velocity. Therefore, energy balance equation can not be solved without considering the Navier-Stokes equations. In the case of a Newtonian incompressible fluid, the governing equations are:

$$\begin{cases} \nabla.(\mathbf{u}) = 0 & \text{(2.4a)} \\ \rho_f \dfrac{\partial \mathbf{u}}{\partial t} + \rho_f\,(\mathbf{u}.\nabla)\mathbf{u} = -\nabla p + \mu\,\nabla.(\nabla \mathbf{u}) + \rho_f\,\mathbf{f} & \text{(2.4b)} \\ \rho_f\,C_{p,f}\dfrac{\partial T_f}{\partial t} + \rho_f\,C_{p,f}\,\mathbf{u}.\nabla(T_f) = k_f\,\nabla.(\nabla T_f) + Q_f & \text{(2.4c)} \end{cases}$$

In the system (2.4), ρ_f stands for the fluid density, μ for the fluid dynamic viscosity, $C_{p,f}$ for the fluid specific heat, **f** for the fluid body forces per unit mass and Q_f is an internal fluid heat source. Q_f will be omitted in our case for two reasons. Firstly, heat dissipation by viscosity in air is insignificant. Secondly, there is no exothermic reaction occuring in the fluid medium. The variables $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, $p = p(\mathbf{x}, t)$ and $T_f = T_f(\mathbf{x}, t)$ are respectively the velocity, the pressure and the temperature field at position **x** and at time t. $(\mathbf{x}, t) \in \Omega_f \times]0, t_e[$ where Ω_f is the fluid domain and $]0, t_e[$ is the time interval for studying the physical phenomenon. In order to take into account the fluid motion created by buoyancy, the energy equation and Navier Stokes equations can be coupled by the Boussinesq approximation [2]:

$$\rho_f(T_f) = \rho_0 \left[1 - \beta(T_f - T_0)\right] \tag{2.5}$$

where ρ_0 is the fluid density at the reference temperature T_0 and β is the fluid thermal expansion coefficient. In such case, equations (2.4) becomes:

$$\begin{cases} \nabla.(\mathbf{u}) = 0 & \text{(2.6a)} \\ \rho_0 \dfrac{\partial \mathbf{u}}{\partial t} + \rho_0 \, (\mathbf{u}.\nabla)\mathbf{u} = -\nabla p + \mu \, \nabla.(\nabla \mathbf{u}) - \rho_0 \, \beta \, (T_f - T_0)\mathbf{g} & \text{(2.6b)} \\ \rho_0 \, C_{p,f} \dfrac{\partial T_f}{\partial t} + \rho_0 \, C_{p,f} \, \mathbf{u}.\nabla(T_f) = k_f \, \nabla.(\nabla T_f) & \text{(2.6c)} \end{cases}$$

where $\mathbf{g} = -g.\mathbf{e}_y$ is the gravity acceleration and $\mathbf{e}_y$ is the y-coordinate unitary vector. Let us define the dimensionless quantitities:

$$\bar{T}_f = \frac{T_f - T_0}{T_w - T_0} \quad \bar{\mathbf{x}} = \frac{\mathbf{x}}{L_c} \quad \bar{\mathbf{u}} = \frac{\mathbf{u}}{u_0} \quad \bar{p} = \frac{p}{\rho_0 . u_0^2} \quad \bar{t} = \frac{t.u_0}{L_c}$$

where T_w is the fluid boundary characteristic temperature, L_c is a characteristic length, u_0 is a characteristic velocity. Therefore, equations (2.6) become:

$$\begin{cases} \nabla.(\bar{\mathbf{u}}) = 0 & \text{(2.7a)} \\ \dfrac{\partial \bar{\mathbf{u}}}{\partial \bar{t}} + (\bar{\mathbf{u}}.\nabla)\bar{\mathbf{u}} = -\nabla \bar{p} + \dfrac{1}{Re} \nabla.(\nabla \bar{\mathbf{u}}) + \dfrac{Gr}{Re^2} \, \bar{T}_f \, \mathbf{e}_y & \text{(2.7b)} \\ \dfrac{\partial \bar{T}_f}{\partial \bar{t}} + \bar{\mathbf{u}}.\nabla \bar{T}_f = \dfrac{1}{Re.Pr} \nabla.(\nabla \bar{T}_f) & \text{(2.7c)} \end{cases}$$

Where $Re = \dfrac{u_0.L_c}{\nu}$, $Gr = \dfrac{g.\beta.(T_w - T_0)L_c^3}{\nu^2}$ and $Pr = \dfrac{\nu}{\alpha}$. ν is the fluid kinematic viscosity and α is the fluid thermal diffusivity. Re, Gr and Pr are respectively the Reynolds, Grashof and Prandtl numbers. An explanation of these numbers is given just below.

Reynolds number

The Reynolds number is defined as:

$$Re = \frac{|inertia\ term|}{|viscous\ term|} = \frac{|\rho_f \, \nabla \mathbf{u}.\mathbf{u}|}{|\mu \, \Delta \mathbf{u}|} = \frac{\rho_f \, u_0 \, L_c}{\mu} \tag{2.8}$$

The Re number allows to determine the flow regime (turbulent or laminar flow). In the case of circular ducts, the critical Reynolds number beyond which the flow becomes turbulent is equal to 2300. When the Reynolds number is high, the viscous term in equation (2.7b) is important only near the solid boundary.

Grashof number
The Grashof number is defined as:

$$Gr = \frac{|buoyancy\ force|}{|viscous\ force|} = \frac{g.\beta.(T_w - T_0)L_c^3}{\nu^2} \tag{2.9}$$

During fluid motion, viscous forces stand against the motion forces generated by buoyancy. Gr number quantifies the ratio of these two effects and allows to judge the type of the convection (forced, natural or mixed). It is illustrated in [1] that:

- When $\frac{Gr}{Re^2} << 1$ the forced convection is dominant
- When $\frac{Gr}{Re^2} >> 1$ the heat transfer is governed by free convection
- When $\frac{Gr}{Re^2} \approx 1$ the combined effect of free and forced convections must be considered.

2.1.3 Conjugate heat transfer (CHT)

Conjugate heat transfer is a problem in which solid and fluid are in contact and exchange heat along the fluid solid interface Γ^w. The scheme 2.1 gives a representation of a thermally coupled fluid-solid problem.

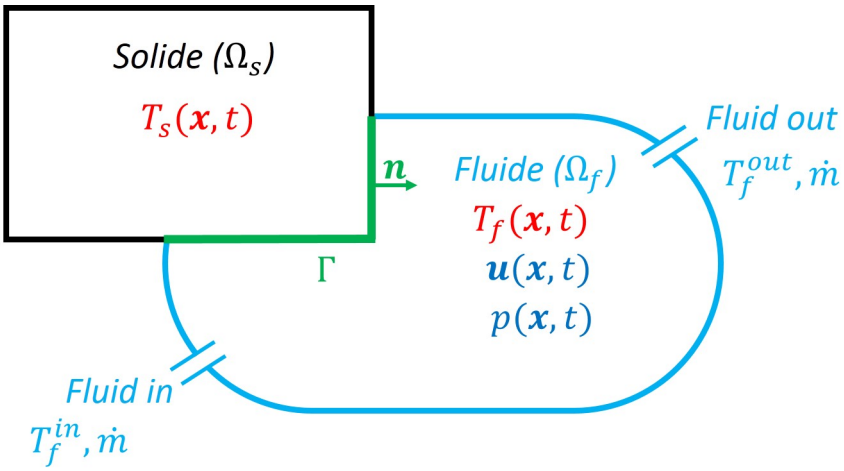


Figure 2.1 – Scheme of conjugate heat transfer

When solving such a problem, the continuity of temperature and heat flux should be verified at the fluid solid interface Γ. These conditions can be written as:

$$\begin{cases} T_f = T_s \quad on\ \Gamma & \text{(2.10a)} \\ q_f = q_s \quad on\ \Gamma & \text{(2.10b)} \end{cases}$$

where $q_s = -k_s \nabla T_s.\mathbf{n}$ and $q_f = k_f \nabla T_f.\mathbf{n}$.

Here $\mathbf{n}$ is the outward pointing normal vector of the solid boundary $\partial\Omega_s$.

2.2 Finite element method

The finite element method (FEM) is a technique for space discretization of partial differential equations (PDE). FEM is based on two main steps:

1. The variational formulation called also weak integral formulation or weighted residual method. In this step a trial solution function is used in the PDE which results in a residual. This residual is minimized over the space domain by means of a weight or test function. Performing a partial integration of the integral (or Green theorem for multivariable functions), decreases the order of derivative in the PDE, which makes differentiability condition less restrictive. That is Why the obtained equation is called weak formulation.

2. The second step consists in approximating the trial solution by many piecewise functions (shape function) over the finite element mesh. Inserting this approximation in the weak formulation and taking weight functions similar to shape function (Galerkin method) yields a system of equations. By solving the system, an approximation of the solution over the mesh is obtained.

A practical illustration of these two main steps will be presented in the case of heat transfer equations in both fluid and solid medium.

2.2.1 Finite element Method for heat transfer in solid medium

The heat transfer problem that will be solved is presented by the following equations:

$$\begin{cases} \rho_s\, C_{p,s}\, \dfrac{\partial T_s(\mathbf{x},t)}{\partial t} - \nabla.(k_s\, \nabla T_s(\mathbf{x},t)) = Q(\mathbf{x},t) & (2.11a) \\ -k_s\, \nabla T_s(\mathbf{x},t).\mathbf{n} = h\, (T_s(\mathbf{x},t) - T_{ext}) \quad on\ \ \partial\Omega_{s1} & (2.11b) \\ -k_s\, \nabla T_s(\mathbf{x},t).\mathbf{n} = 0 \quad on\ \ \partial\Omega_{s2} & (2.11c) \\ T_s(\mathbf{x},t=0) = T_{s0}(\mathbf{x}) & (2.11d) \end{cases}$$

where $\partial\Omega_{s1}$ is a part of the solid boundary submitted to convective heat exchange and $\partial\Omega_{s2}$ is a part of the solid boundary which is insolated. In this case we consider two partitions of the solid boundary : $\partial\Omega_{s1} \cup \partial\Omega_{s2} = \partial\Omega_s$ but the methodology remains the same when treating a case consisting of many partitions for the solid boundary.

2.2.1.1 Weighted integral formulation

Lets denote by $\hat{T}_s(\mathbf{x},t)$ a trial solution function. The aim of the FEM is to compute $\hat{T}_s(\mathbf{x},t)$. The residual resulting from using the trial function in equations (2.11a) is:

$$R(\hat{T}_s(\mathbf{x},t)) = \rho_s\, C_{p,s}\, \frac{\partial \hat{T}_s(\mathbf{x},t)}{\partial t} - \nabla.(k_s\, \nabla \hat{T}_s(\mathbf{x},t)) - Q(\mathbf{x},t) \tag{2.12}$$

The weighted residual method consists in canceling the residual by means of a weight function $T_s^*(\mathbf{x})$ over the space domain Ω_s. This can be written as :

$$\int_{\Omega_s} R(\hat{T}_s(\mathbf{x},t))\; T_s^*(\mathbf{x})\; d\Omega_s = 0 \tag{2.13}$$

Using the Green theorem (Appendix C) allows to obtain the variational formulation given by equation (2.14):

$$\int_{\Omega_s} \rho_s \, C_{p,s} \, \frac{\partial \hat{T}_s(\mathbf{x},t)}{\partial t} \, T_s^*(\mathbf{x}) \, d\Omega_s + \int_{\Omega_s} k_s \, \nabla \hat{T}_s . \nabla T_s^* \, d\Omega_s = \int_{\Omega_s} Q(\mathbf{x},t) \, T_s^*(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, T_s^* \, d(\partial\Omega_s) \tag{2.14}$$

where $\mathbf{q}_n = -k_s \nabla \hat{T}_s . \mathbf{n}$ is the normal conduction heat flux on the solid boundary $\partial\Omega_{s1}$. Calculation details are given in Appendix B.

2.2.1.2 Meshing and discrete weak formulation

When solving numericaly the heat equation, the temperature will be determined in a set of positions (or nodes). These nodes are the result of the domain discretization into elements. Each element contains many nodes. For instance, the finite element mesh of one-leg power module is presented in Figure 2.2.

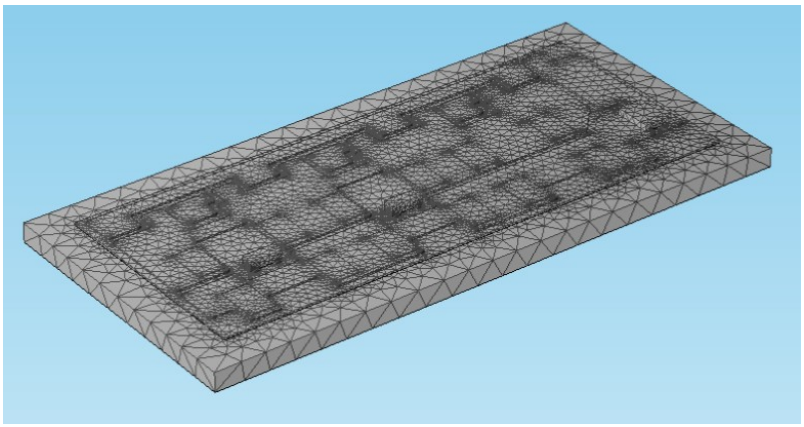


Figure 2.2 – Mesh of one-leg power module

The temperature field $\hat{T}_s(\mathbf{x},t)$ in the system domain can be approximated using shape functions. This approximation is given by:

$$\hat{T}_s(\mathbf{x},t) = \sum_{j=1}^{M_{TS}} \chi_{s,j}(\mathbf{x}) \, T_{s,j}(t) \tag{2.15}$$

where $\{\chi_{s,j}(\mathbf{x})\}_{1 \le j \le M_{TS}}$ are the shape functions, $\{T_{s,j}(\mathbf{x})\}_{1 \le j \le M_{TS}}$ are the nodal temperature and M_{TS} is the number of temperature nodes in the solid domain.

Various kind of shape functions can be used : linear, quadratic cubic, etc. Shape functions are usually obtained by Lagrange interpolation. Calculation methods and analytical expressions of these shape functions can be found in [3,4]. The order of interpolation is related to the number of nodes per element.

It is obvious that when the number of nodes per element increases (the order increase), the quality of the temperature approximation improves and consequently the temperature is computed with a higher accuracy.

Inserting the finite element approximation in equation (2.14) and choosing weight functions identical to shape functions (see Appendix B) yields equation (2.16):

$$\sum_{j=1}^{M_{TS}} \left(\int_{\Omega_s} \rho_s \, C_{p,s} \, \chi_{s,j}(\mathbf{x}) \, \chi_{s,i}(\mathbf{x}) \; d\Omega_s \right) \frac{dT_{s,j}(t)}{dt} + \sum_{j=1}^{M_{TS}} \left(\int_{\Omega_s} k_s \, \nabla\chi_{s,j}(\mathbf{x}).\nabla\chi_{s,i}(\mathbf{x}) \, d\Omega_s \right) T_{s,j}(t) = \int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, \chi_{s,i}(\mathbf{x}) \, d(\partial\Omega_s) \quad for \;\; i = 1, \ldots, M_{TS} \tag{2.16}$$

Let's define:

$$\begin{cases} C_{ij} = \displaystyle\int_{\Omega_s} \rho_s \, C_{p,s} \, \chi_{s,i}(\mathbf{x}) \, \chi_{s,j}(\mathbf{x}) \, d\Omega_s \quad the \;\; mass \;\; matrix & \text{(2.17a)} \\ K_{ij} = \displaystyle\int_{\Omega_s} k_s \, \nabla\chi_{s,i}(\mathbf{x}) \, \nabla\chi_{s,j}(\mathbf{x}) \, d\Omega_s \quad the \;\; stiffness \;\; matrix & \text{(2.17b)} \\ l_i(t) = \displaystyle\int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, \chi_{s,i}(\mathbf{x}) \; d(\partial\Omega_s) \quad the \;\; load \;\; vector & \text{(2.17c)} \end{cases}$$

The finite element model can be written as :

$$\sum_{j=1}^{M_{TS}} C_{ij} \, \frac{dT_{s,j}(t)}{dt} + \sum_{j=1}^{M_{TS}} K_{ij} \, T_{s,j}(t) = l_i(t) \quad for \;\; i = 1, \ldots, M_{TS} \tag{2.18}$$

Equations (2.18) represent a $M_{TS} \times M_{TS}$ linear system for the M_{TS} unknowns $T_{s,j}$. In matrix form, equation (2.18) can be written as:

$$\mathbf{C} \, \frac{d\mathbf{T}_s(t)}{dt} + \mathbf{K} \, \mathbf{T}_s(t) = \mathbf{l}(t) \tag{2.19}$$

where $\mathbf{C} = (C_{ij})_{1 \leq i,j \leq M_{TS}}$ is the mass matrix and is also called heat capacity matrix. $\mathbf{K} = (K_{ij})_{1 \leq i,j \leq M_{TS}}$ is the stiffness matrix and is also called heat conductivity matrix. $\mathbf{l} = (l_i)_{1 \leq i \leq M_{TS}}$ is the load vector that takes into account the heat source contribution and the boundary conditions. $\mathbf{T}_s(t)$ is the nodal temperature vector. It can be seen from equations (2.17a) and equations (2.17b) that both the capacity and thermal conductivity matrix are symmetric matrices.

A necessary requirement for the existence of integrals in equations (2.18) is that the shape functions and their derivatives are square integrable functions over Ω_s. This means that the shape function used to approximate the temperature field are belonging to a Sobolev space of order 0 [5].

When inserting the expression of q_n given by equation (2.11b) in equation (2.17c), the load vector becomes:

$$l_i(t) = \int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} h \, (\hat{T}_s(\mathbf{x},t) - T_{ext}) \, \chi_{s,i}(\mathbf{x}) \, d(\partial\Omega_s) \tag{2.20}$$

Using the approximation of the temperature given by equation (2.15) in equation (2.20) yields:

$$\begin{aligned} l_i(t) &= \int_{\Omega_s} Q(\mathbf{x},t)\ \chi_{s,i}(\mathbf{x})\ d\Omega_s - \int_{\partial\Omega_{s1}} h \left(\sum_{j=1}^{M_{TS}} \chi_{s,j}(\mathbf{x})\ T_{s,j}(t) - T_{ext} \right) \chi_{s,i}(\mathbf{x})\ d(\partial\Omega_s) \\ &= \int_{\Omega_s} Q(\mathbf{x},t)\ \chi_{s,i}(\mathbf{x})\ d\Omega_s + h\ T_{ext} \int_{\partial\Omega_{s1}} \chi_{s,i}(\mathbf{x})\ d(\partial\Omega_s) \\ &- h \sum_{j=1}^{M_{TS}} \left(\int_{\partial\Omega_{s1}} \chi_{s,j}(\mathbf{x})\ \chi_{s,i}(\mathbf{x})\ d(\partial\Omega_s) \right) T_{s,j}(t) \end{aligned} \tag{2.21}$$

The last term in equation (2.21) is a linear function of the nodal temperature $T_{s,j}(t)$. Therefore, this term should be put with the stiffness matrix. In such case, the stiffness matrix becomes:

$$\begin{aligned} K_{ij} &= \int_{\Omega_s} k_s\ \nabla\chi_{s,i}(\mathbf{x})\ \nabla\chi_j(\mathbf{x})\ d\Omega_s + h \int_{\partial\Omega_{s1}} \nabla\chi_{s,i}(\mathbf{x})\ \nabla\chi_{s,j}(\mathbf{x})\ d\Omega_s \\ &= K_{ij,0} + h.K_{ij,cv} \end{aligned} \tag{2.22}$$

Consequently, the load vector becomes:

$$\begin{aligned} l_i(t) &= \int_{\Omega_s} Q(\mathbf{x},t)\ \chi_{s,i}(\mathbf{x})\ d\Omega_s + h\ T_{ext} \int_{\partial\Omega_{s1}} \chi_{s,i}(\mathbf{x})\ d(\partial\Omega_s) \\ &= l_{i,Q}(t) + h.T_{ext}.l_{i,cv} \end{aligned} \tag{2.23}$$

In conclusion, the matrix form of the finite element model can be written as:

$$\mathbf{C}\ \frac{d\mathbf{T}_s(t)}{dt} + (\mathbf{K}_0 + h\ \mathbf{K}_{cv})\ \mathbf{T}_s(t) = \mathbf{l}_Q(t) + h\ T_{ext}\ \mathbf{l}_{cv} \tag{2.24}$$

where $\mathbf{K}_0$ is the matrix having the coefficients $\int_{\Omega_s} k_s\ \nabla\chi_{s,i}(\mathbf{x})\ \nabla\chi_{s,j}(\mathbf{x})\ d\Omega_s$, $\mathbf{K}_{cv}$ is the matrix having the coefficients $\int_{\partial\Omega_{s1}} \nabla\chi_{s,i}(\mathbf{x})\ \nabla\chi_{s,j}(\mathbf{x})\ d\Omega_s$, $\mathbf{l}_Q(t)$ is the load vector corresponding to the heat source and having the coefficients $\int_{\Omega_s} Q(\mathbf{x},t)\ \chi_{s,i}(\mathbf{x})\ d\Omega_s$ and $\mathbf{l}_{cv}$ is the load vector corresponding to the convection boundary condition and having the coefficients $\int_{\partial\Omega_{s1}} \chi_{s,i}(\mathbf{x})\ d(\partial\Omega_s)$. All the previous matrices are recovered from the industrial software $COMSOL^{®}$ and have been used to construct the direct reduced order model.

2.2.1.3 Analytical solution

Equation (2.19) can be written in a different form:

$$\frac{d\mathbf{T}_s(t)}{dt} = -\mathbf{C}^{-1}\ \mathbf{K}\ \mathbf{T}_s(t) + \mathbf{C}^{-1}\ \mathbf{l}(t) \tag{2.25}$$

Let's define:

$$\begin{cases} \mathbf{A} = -\mathbf{C}^{-1}\ \mathbf{K} \\ \mathbf{b}(t) = \mathbf{C}^{-1}\ \mathbf{l}(t) \end{cases}$$

Then, equation (2.25) becomes:

$$\frac{d\mathbf{T}_s(t)}{dt} = \mathbf{A}\ \mathbf{T}_s(t) + \mathbf{b}(t) \tag{2.27}$$

and has an analytical solution (Appendix D) that can be written as:

$$\mathbf{T}_s(t) = \exp(\mathbf{A}\ t)\ \mathbf{T}_{s0} + \int_0^t \exp(\mathbf{A}\ (t-\tau))\mathbf{b}(\tau)\mathrm{d}\tau \tag{2.28}$$

where the exponential of a matrix is defined as:

$$\exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \tag{2.29}$$

It is obvious that computing the exponential of a matrix is time consuming as it involves computing the power of a matrix at each time t. In order to overcome this inconvenience, time discretization is performed as it will be depicted in the following section.

2.2.1.4 Temporal discretization

Equation (2.19) represents an approximation of the PDE (2.11a) which is discrete in space and continuous in time. The aim of time integration is to obtain the time evolution of temperature at the mesh nodes.

For this purpose, a discrete time integration method will be applied on equation (2.19). The time interval $[0, t_e]$ will be subdivided into N_t time intervals of length $\Delta t = t_{n+1} - t_n = t_n - t_{n-1}$. The outcome of time integration is called the numerical scheme. Many numerical schemes exist and are listed in Appendix B.

In our case, the numerical scheme that has been chosen is the Crank Nicolson scheme as it gathers high-order accuracy and low number of mathematical operations as shown in Appendix B. This numerical scheme is:

$$\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)\mathbf{T}_s^{n+1} = \left(\frac{1}{\Delta t}\mathbf{C} - \frac{1}{2}\mathbf{K}\right)\mathbf{T}_s^{n} + \frac{1}{2}\left(\mathbf{l}^{n+1} + \mathbf{l}^{n}\right) \tag{2.30}$$

Once the matrix $\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)$ is invertible, equation (2.30) can be written in a practical format:

$$\mathbf{T}_s^{n+1} = \left(\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)^{-1}\left(\frac{1}{\Delta t}\mathbf{C} - \frac{1}{2}\mathbf{K}\right)\right).\mathbf{T}_s^{n} + \left(\frac{1}{2}\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)^{-1}\right).\left(\mathbf{l}^{n+1} + \mathbf{l}^{n}\right) \tag{2.31}$$

This method is inconditionally stable and second-order accurate in time.

2.2.2 Finite element Method for heat transfer in fluid medium

In this section, equations (2.6) will be discretized using the finite element method.

2.2.2.1 Weighted integral formulation

Using the trial solution $\hat{\mathbf{u}}(\mathbf{x},t)$, $\hat{p}(\mathbf{x},t)$ and $\hat{T_f}(\mathbf{x},t)$ in equations (2.6) yields the following residuals:

$$\begin{cases} R_1(\hat{\mathbf{u}}) = \nabla.(\hat{\mathbf{u}}) & \text{(2.32a)} \\ R_2(\hat{\mathbf{u}},\hat{p},\hat{T_f}) = \rho_0 \dfrac{\partial \hat{\mathbf{u}}}{\partial t} + \rho_0(\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}} + \nabla\hat{p} - \mu\nabla.(\nabla\hat{\mathbf{u}}) + \rho_0\,\beta\,(\hat{T_f}-T_0)\,\mathbf{g} & \text{(2.32b)} \\ R_3(\hat{T_f},\hat{\mathbf{u}}) = \rho_0\,C_{p,f}\dfrac{\partial \hat{T_f}}{\partial t} + \rho_0\,C_{p,f}\,\hat{\mathbf{u}}.\nabla(\hat{T_f}) - \nabla.(k_f\,\nabla\hat{T_f}) & \text{(2.32c)} \end{cases}$$

The weighted residual method consists in canceling all the residuals using the weight functions $p^*(\mathbf{x})$, $\mathbf{u}^*(\mathbf{x})$ and $T_f^*(\mathbf{x})$. This can be written as:

$$\begin{cases} \displaystyle\int_{\Omega_f} R_1(\hat{\mathbf{u}}).p^*.d\Omega_f = 0 & \text{(2.33a)} \\ \displaystyle\int_{\Omega_f} R_2(\hat{\mathbf{u}},\hat{p},\hat{T_f}).\mathbf{u}^*.d\Omega_f = 0 & \text{(2.33b)} \\ \displaystyle\int_{\Omega_f} R_3(\hat{T_f},\hat{\mathbf{u}}).\hat{T_f}^*.d\Omega_f = 0 & \text{(2.33c)} \end{cases}$$

which yields

$$\begin{cases} \displaystyle\int_{\Omega_f} \nabla.(\hat{\mathbf{u}})\,p^*\,d\Omega_f = 0 & \text{(2.34a)} \\ \displaystyle\int_{\Omega_f}\left(\rho_0\frac{\partial\hat{\mathbf{u}}}{\partial t}.\mathbf{u}^* + \rho_0(\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}}.\mathbf{u}^* + \nabla\hat{p}.\mathbf{u}^* - \mu\,\Delta\hat{\mathbf{u}}.\mathbf{u}^* + \rho_0\,\beta\,(\hat{T_f}-T_0)\,\mathbf{g}.\mathbf{u}^*\right)d\Omega_f = 0 & \text{(2.34b)} \\ \displaystyle\int_{\Omega_f}\rho_0\,C_{p,f}\frac{\partial\hat{T_f}}{\partial t}\,T_f^* + \rho_0\,C_{p,f}\,\hat{\mathbf{u}}.\nabla(\hat{T_f})\,T_f^* - k_f\,\Delta\hat{T_f}\,T_f^*\,d\Omega_f = 0 & \text{(2.34c)} \end{cases}$$

The terms $\displaystyle\int_{\Omega_f}\nabla\hat{p}.\mathbf{u}^*\,d\Omega_f$, $\displaystyle\int_{\Omega_f}\Delta\hat{\mathbf{u}}.\mathbf{u}^*\,d\Omega_f$ and $\displaystyle\int_{\Omega_f}\Delta\hat{T_f}\,T_f^*\,d\Omega_f$ can be developped using Green's theorem as depicted in Appendix B.

Therefore, the variational formulation given by equations (2.34) can be written as:

$$\begin{cases} a_0(\hat{\mathbf{u}},p^*) = 0 & \text{(2.35a)} \\ \left(\rho_0\dfrac{\partial\hat{\mathbf{u}}}{\partial t},\mathbf{u}^*\right) + n_u(\hat{\mathbf{u}},\hat{\mathbf{u}},\mathbf{u}^*) + a_u(\hat{\mathbf{u}},\mathbf{u}^*) - a_0(\mathbf{u}^*,\hat{p}) = -b(\hat{T_f},\mathbf{u}^*) & \text{(2.35b)} \\ \left(\rho_0\,C_{p,f}\dfrac{\partial\hat{T_f}}{\partial t},T_f^*\right) + n_T(\hat{T_f},\hat{\mathbf{u}},T_f^*) + a_T(\hat{T_f},T_f^*) = -c(q^w,T_f^*) & \text{(2.35c)} \end{cases}$$

where

$$\begin{cases} a_u(\hat{\mathbf{u}}, \mathbf{u}^*) = \int_{\Omega_f} \mu \ \nabla\hat{\mathbf{u}}.\nabla\mathbf{u}^* \ d\Omega_f & (2.36a) \\ a_T(\hat{T_f}, T_f^*) = \int_{\Omega_f} k_f \ \nabla\hat{T_f}.\nabla T_f^* \ d\Omega_f & (2.36b) \\ a_0(\hat{\mathbf{u}}, p^*) = \int_{\Omega_f} \nabla.(\hat{\mathbf{u}}) \ p^* \ d\Omega_f & (2.36c) \\ b(\hat{T_f}, \mathbf{u}^*) = \int_{\Omega_f} \rho_0 \ \beta \ (\hat{T_f} - T_0)\mathbf{g}.\mathbf{u}^* \ d\Omega_f & (2.36d) \\ n_u(\mathbf{u}, \mathbf{v}, \mathbf{u}^*) = \int_{\Omega_f} \rho_0(\mathbf{u}.\nabla)\mathbf{v}.\mathbf{u}^* \ d\Omega_f & (2.36e) \\ n_T(\hat{T_f}, \hat{\mathbf{u}}, T_f^*) = \int_{\Omega_f} \rho_0 \ C_{p,f} \ \hat{\mathbf{u}}.\nabla(\hat{T_f}) \ T_f^* \ d\Omega_f & (2.36f) \\ c(q^w, T_f^*) = \int_{\partial\Omega_f} q^w(\mathbf{x}) \ T_f^* \ d(\partial\Omega_f) & (2.36g) \\ & (2.36h) \end{cases}$$

and the scalar product for two arbitrary vectors **a** and **b** is defined as follows:

$$(\mathbf{a}, \mathbf{b}) = \int_{\Omega_f} \mathbf{a}.\mathbf{b} \ d\Omega_f \tag{2.37}$$

The requirement for the existence of the bilinear, trilinear forms (equations (2.36)) used in the weak formulation is that:

- $\hat{p}(\mathbf{x}), \ p^*(\mathbf{x}) \ \in \ H^0(\Omega_f)$
- $\hat{T_f}(\mathbf{x}), \ T_f^*(\mathbf{x}), \ \hat{\mathbf{u}}(\mathbf{x}), \ \mathbf{u}^*(\mathbf{x}) \ \in \ H^1(\Omega_f)$

where $H^0(\Omega_f)$ and $H^1(\Omega_f)$ are respectively Sobolev spaces of order 0 and 1.

2.2.2.2 Meshing and discrete weak formulation

The problem unknowns are $\hat{\mathbf{u}}(\mathbf{x}, t)$, $\hat{p}(\mathbf{x}, t)$ and $\hat{T_f}(\mathbf{x}, t)$. These unknowns will be approximated using shape functions:

$$\begin{cases} \hat{\mathbf{u}}(\mathbf{x}, t) = \sum_{j=1}^{M_u} \boldsymbol{\sigma_j}(\mathbf{x}).u_j(t) & (2.38a) \\ \hat{p}(\mathbf{x}, t) = \sum_{j=1}^{M_p} \beta_i(\mathbf{x}).p_j(t) & (2.38b) \\ \hat{T_f}(\mathbf{x}, t) = \sum_{j=1}^{M_{Tf}} \chi_{f,j}(\mathbf{x}).T_{f,j}(t) & (2.38c) \end{cases}$$

where M_p and M_{Tf} are respectively the number of pressure and temperature nodes in the fluid medium. M_u is the number of velocity degrees of freedom. M_u is three times the number of velocity nodes when dealing with 3-dimensional fluid dynamic problem.

$u_j(t)$, $p_j(t)$ and $T_{f,j}(t)$ are respectively nodal (at nodes) values of velocity, pressure and temperature at time t. An important rule when choosing shape functions for both velocity and pressure is that the interpolation used for pressure must be at least one order lower than that used for the velocity field. This rule aims to prevent an overconstrained system of discrete equations [6].

The Galerkin method consists in substituting the weight functions $\mathbf{u}^*(\mathbf{x})$, $p^*(\mathbf{x})$ and $T_f^*(\mathbf{x})$ by the corresponding shape function $\boldsymbol{\sigma_i}(\mathbf{x})$, $\beta_i(\mathbf{x})$ and $\chi_{f,i}(\mathbf{x})$. $\beta_i(\mathbf{x})$ and $\chi_{f,i}(\mathbf{x})$ are scalar functions whereas $\boldsymbol{\sigma_i}(\mathbf{x})$ is a vector function belonging to $\mathbb{R}^3$ in the case of a three dimensional velocity field. The variational formulation can be written as:

$$\begin{cases} a_0(\hat{\mathbf{u}}, \beta_i(\mathbf{x})) = 0 \quad for \quad i = 1, \dots, M_p & \text{(2.39a)} \\ \left(\rho_0 . \dfrac{\partial \hat{\mathbf{u}}}{\partial t}, \boldsymbol{\sigma_i}(\mathbf{x})\right) + n_u(\hat{\mathbf{u}}, \hat{\mathbf{u}}, \boldsymbol{\sigma_i}(\mathbf{x})) + a_u(\hat{\mathbf{u}}, \boldsymbol{\sigma_i}(\mathbf{x})) & \\ \qquad - a_0(\boldsymbol{\sigma_i}(\mathbf{x}), \hat{p}) = -b(\hat{T_f}, \boldsymbol{\sigma_i}(\mathbf{x})) \quad for \quad i = 1, \dots, M_u & \text{(2.39b)} \\ \left(\rho_0 . C_{p,f} \dfrac{\partial \hat{T_f}}{\partial t}, \chi_{f,i}\right) + n_T(\hat{T_f}, \hat{\mathbf{u}}, \chi_{f,i}) & \\ \qquad + a_T(\hat{T_f}, \chi_{f,i}) = -c(q^w, \chi_{f,i}) \quad for \quad i = 1, \dots, M_{Tf} & \text{(2.39c)} \end{cases}$$

Replacing the finite element approximation of the velocity, pressure and temperature field (equations (2.38)) in the variational formulation (equations (2.39)) yields :

$$\begin{cases} \displaystyle\sum_{j=1}^{M_u} \left(\mathbf{L}_f\right)_{ij} . u_j(t) = 0 \quad for \quad i = 1, \dots, M_p & \text{(2.40a)} \\ \displaystyle\sum_{j=1}^{M_u} \left(\mathbf{C}_{fu}\right)_{ij} . \frac{du_j(t)}{dt} + \sum_{j=1}^{M_u} \left(\left(\mathbf{K}_{fu}\right)_{ij} + \left(\mathbf{N}_{fu}(\hat{\mathbf{u}})\right)_{ij} \right) . u_j(t) - \sum_{j=1}^{M_p} \left(\mathbf{L}_f\right)_{ij} . p_j(t) & \\ \qquad = - \displaystyle\sum_{j=1}^{M_{Tf}} \left(\mathbf{B}_{fu}\right)_{ij} . T_{f,j}(t) + \mathbf{d}_i \quad for \quad i = 1, \dots, M_u & \text{(2.40b)} \\ \displaystyle\sum_{j=1}^{M_{Tf}} \left(\mathbf{C}_{fT}\right)_{ij} . \frac{dT_{f,j}(t)}{dt} + \sum_{j=1}^{M_{Tf}} \left(\left(\mathbf{K}_{fT}\right)_{ij} + \left(\mathbf{N}_{fT}(\hat{\mathbf{u}})\right)_{ij} \right) . T_{f,j}(t) & \\ \qquad = - \left(\mathbf{Q}_f\right)_i \quad for \quad i = 1, \dots, M_{Tf} & \text{(2.40c)} \end{cases}$$

where

$$
\begin{cases}
(\mathbf{L}_f)_{ij} = a_0\left(\boldsymbol{\sigma_j}(\mathbf{x}), \beta_i(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_p \;\; and \;\; 1 \leq j \leq M_u & (2.41a)\\
(\mathbf{C}_{fu})_{ij} = \left(\rho_0\, \boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_u & (2.41b)\\
(\mathbf{K}_{fu})_{ij} = a_u\left(\boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_u & (2.41c)\\
(\mathbf{N}_{fu})_{ij} = n_u\left(\sum_{k=1}^{M_u} \boldsymbol{\sigma_k}(\mathbf{x})\, u_k(t), \boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_u & (2.41d)\\
(\mathbf{B}_{fu})_{ij} = \left(\rho_0\, \beta\, \chi_{f,j}(\mathbf{x})\, \mathbf{g}, \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq j \leq M_{Tf} \;\; and \;\; 1 \leq i \leq M_u & (2.41e)\\
(\mathbf{d})_i = \left(\rho_0\, \beta\, T_0\, \mathbf{g}, \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_u & (2.41f)\\
(\mathbf{C}_{fT})_{ij} = \left(\rho_0\, C_{p,f}\, \chi_{f,j}(\mathbf{x}), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_{Tf} & (2.41g)\\
(\mathbf{K}_{fT})_{ij} = a_T\left(\chi_{f,j}(\mathbf{x}), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_{Tf} & (2.41h)\\
(\mathbf{N}_{fT})_{ij} = n_T\left(\chi_{f,j}(\mathbf{x}), \sum_{k=1}^{M_u} \boldsymbol{\sigma_k}(\mathbf{x})\, u_k(t), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i,j \leq M_{Tf} & (2.41i)\\
\left(\mathbf{Q}_f\right)_i = \left(q^w, \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_{Tf} & (2.41j)\\
& (2.41k)
\end{cases}
$$

Calculation details can be found in Appendix B.
Equations (2.40) can be cast in the following matrix form:

$$\mathbf{C}.\frac{d\mathbf{X}(t)}{dt} + \mathbf{K}(\mathbf{U}).\mathbf{X}(t) = \mathbf{L} \tag{2.42}$$

where

$$\mathbf{X}(t) = \left(\mathbf{P}(t), \mathbf{U}(t), \mathbf{T}_f(t)\right)^T$$

$$\mathbf{U}(t) = \left(u_1(t), u_2(t), \ldots, u_{M_u}(t)\right)$$

$$\mathbf{P}(t) = \left(p_1(t), \ldots, p_{M_p}(t)\right)$$

$$\mathbf{T}_f(t) = \left(T_{f,1}(t), \ldots, T_{f,M_{Tf}}\right)$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{fu} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{fT} \end{pmatrix}$$

$$\mathbf{K}(\mathbf{U}) = \begin{pmatrix} \mathbf{0} & \mathbf{L}_f & \mathbf{0} \\ -\mathbf{L}_f^T & \mathbf{K}_{fu} + \mathbf{N}_{fu}(\mathbf{U}) & \mathbf{B}_f^T \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{fT} + \mathbf{N}_{fT}(\mathbf{U}) \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \\ -\mathbf{Q}_f \end{pmatrix}$$

Equations (2.42) are non-linear because the stiffness matrix depends on the velocity. Many numerical methods can be used to solve equation (2.42) such as the fixed point algorithm or Newton Raphson method. These methods are well described in [6] for the case of heat transfer and fluid dynamics equations.

2.2.3 Finite Element Model for CHT

Conjugate heat transfer (CHT) involves simultaneous heat transfer in fluid and solid medium. In this section, the coupling between the finite element model in solid and fluid medium will be described.

The coupling occuring at the interface between the solid and the fluid is quantified by equations (2.10) stating the continuity of heat flux and temperature at the fluid-solid interface.

In practice, one variable is used as boundary condition (heat flux continuity), where the second variable is used to measure the solution convergence (temperature continuity) [6].

For the sake of simplicity, the heat exchange at the fluid-solid interface will be modeled by a convective heat transfer coefficient h.

The second term of $l_i(t)$ (equation (2.17c)) is responsible for the coupling with heat transfer in the fluid medium. This term takes into account the heat transfer across Ω_{s1} corresponding to the fluid-solid interface. To alleviate the notation this surface will be denoted by Γ. Let's define the corresponding bilinear form $(\ .\ ,\ .\ )_\Gamma$ as :

$$Q_{s,i} = (q_n, \chi_{s,i}(\mathbf{x}))_\Gamma = -\int_\Gamma q_n . \chi_{s,i}(\mathbf{x}).\ d\Gamma \tag{2.43}$$

where $q_n = -k_s \nabla \hat{T}_s.\mathbf{n} = h(\hat{T}_s(\mathbf{x},t) - \hat{T}_f(\mathbf{x},t))\ \ \mathbf{x} \in \Gamma$.

Therefore:

$$Q_{s,i} = \left(h(\hat{T}_s(\mathbf{x},t) - \hat{T}_f(\mathbf{x},t), \chi_{s,i}(\mathbf{x})\right)_\Gamma \tag{2.44}$$

$$= h.\left(\hat{T}_s(\mathbf{x},t), \chi_{s,i}(\mathbf{x})\right)_\Gamma - h.\left(\hat{T}_f(\mathbf{x},t), \chi_{s,i}(\mathbf{x})\right)_\Gamma \tag{2.45}$$

Inserting the finite element approximation of the fluid and solid temperature in equation (2.45) yields:

$$\begin{aligned}
Q_{s,i} &= h.\left(\hat{T}_s(\mathbf{x},t), \chi_{s,i}(\mathbf{x})\right)_\Gamma - h.\left(\hat{T}_f(\mathbf{x},t), \chi_{s,i}(\mathbf{x})\right)_\Gamma \\
&= h.\left(\sum_{j=1}^{M_{TS}} \chi_{s,j}(\mathbf{x})\ T_{s,j}(t), \chi_{s,i}(\mathbf{x})\right)_\Gamma - h.\left(\sum_{j=1}^{M_{Tf}} \chi_{f,j}(\mathbf{x})\ T_{f,j}(t), \chi_{s,i}(\mathbf{x})\right)_\Gamma \\
&= \sum_{j=1}^{M_{TS}} (h.\chi_{s,j}(\mathbf{x})\ , \chi_{s,i}(\mathbf{x}))_\Gamma .T_{s,j}(t) - \sum_{j=1}^{M_{Tf}} (h.\chi_{f,j}(\mathbf{x})\ , \chi_{s,i}(\mathbf{x}))_\Gamma .T_{f,j}(t) \\
&= \sum_{j=1}^{M_{TS}} (K_{ss})_{ij} .T_{s,j}(t) - \sum_{j=1}^{M_{Tf}} (K_{fs})_{ij} .T_{f,j}(t)
\end{aligned} \tag{2.46}$$

Similarly, in the fluid medium the coupling term is :

$$Q_{f,i} = (q_f, \chi_{f,i}(\mathbf{x}))_\Gamma \tag{2.47}$$

where $q_f = -k_f \nabla \hat{T}_f.(\textbf{-n}) = k_f \nabla \hat{T}_f.\mathbf{n} = h(\hat{T}_f(\mathbf{x},t) - \hat{T}_s(\mathbf{x},t)) \;\; \mathbf{x} \in \Gamma$.

Using the finite element approximation and developping as in equation (2.46) yields:

$$\begin{aligned} Q_{f,i} &= h.\left(\hat{T}_f(\mathbf{x},t), \chi_{f,i}(\mathbf{x})\right)_\Gamma - h.\left(\hat{T}_s(\mathbf{x},t), \chi_{f,i}(\mathbf{x})\right)_\Gamma \\ &= \sum_{j=1}^{M_{Tf}} \left(h.\chi_{f,j}(\mathbf{x}) \, , \chi_{f,i}(\mathbf{x})\right)_\Gamma .T_{f,j}(t) - \sum_{j=1}^{M_{Ts}} \left(h.\chi_{s,j}(\mathbf{x}) \, , \chi_{f,i}(\mathbf{x})\right)_\Gamma .T_{s,j}(t) \\ &= \sum_{j=1}^{M_{Tf}} \left(K_{ff}\right)_{ij} .T_{f,j}(t) - \sum_{j=1}^{M_{Ts}} \left(K_{sf}\right)_{ij} .T_{s,j}(t) \end{aligned} \tag{2.48}$$

Equations (2.46) and (2.48) can be cast in the following matrix form:

$$\begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_f \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{ss} & \mathbf{K}_{fs} \\ \mathbf{K}_{sf} & \mathbf{K}_{ff} \end{pmatrix} \begin{pmatrix} \mathbf{T}_s \\ \mathbf{T}_f \end{pmatrix} \tag{2.49}$$

The conjugate heat transfer finite element model consists in solving equations (2.40) and (2.18) in combination with equations (2.49).

2.3 Used finite element software: Comsol Multiphysics

Comsol Multiphysics has been used to run the finite element simulations. This choice is made for many reasons:

- Comsol can be linked to $MATLAB^{®}$ via the livelink application. This link allows to recover finite element matrices, the solution and the mesh on $MATLAB^{®}$. This is convenient to test many time integration schemes or various operating conditions. The $MATLAB^{®}$ library containing all these functions is called mli. For instance, the most handy functions of the mli library are:
  - **mpheval** : evaluates expressions on node points
  - **mphgetu** : returns solution vector
  - **mphmatrix** : returns finite element model matrices
  - **mphplot** : renders a plot group in a figure window

  Details about mli functions are depicted in [7].
- Comsol is a multiphysics software in which adding physics can be implemented easily compared to other softwares.
- Comsol uses the finite element method which is effective when dealing with complex geometries. This is the case for I2MPECT power electronic module in which many geometric details exist.

The main reason for using $COMSOL^{®}$ is the possibility to recover finite element matrices. Finite element matrices are required to construct the reduced order model as it will be depicted in Chapter 3.

The first step to study heat transfer in the power module is to construct the finite element model in the $COMSOL^{®}$ **framework**. During this step, the geometry of the power module is imported (CAD file) or constructed. Material properties of each part are specified. Then boundary and intial conditions are chosen. After that, the mesh is constructed and the discretization order is chosen (shape functions). The solution can then be computed by the solver in which some parameters are required such as time interval, time step, time discretization method. The model should then be saved.

The second step is to open the same model using LiveLink™ for $MATLAB^{®}$. This procedure opens automatically matlab with the mli library. By using the function **mphmatrix** of the mli library, matrices **C** , **K** and **l** are recovered. These matrices are saved as matlab file so a new solution for new operating conditions can be computed in $MATLAB^{®}$. To compute a new solution, the Crank-Nicolson discretization scheme is used. This procedure has been used on test cases and obtained results show good agreement between Comsol results and those computed on $MATLAB^{®}$.

A summary of this procedure is presented on Figure 2.3

- **Remark:**

  $COMSOL^{®}$ is only used to construct the finite element matrices. Henceforth, the time integration of the finite element model is performed using $MATLAB^{®}$.

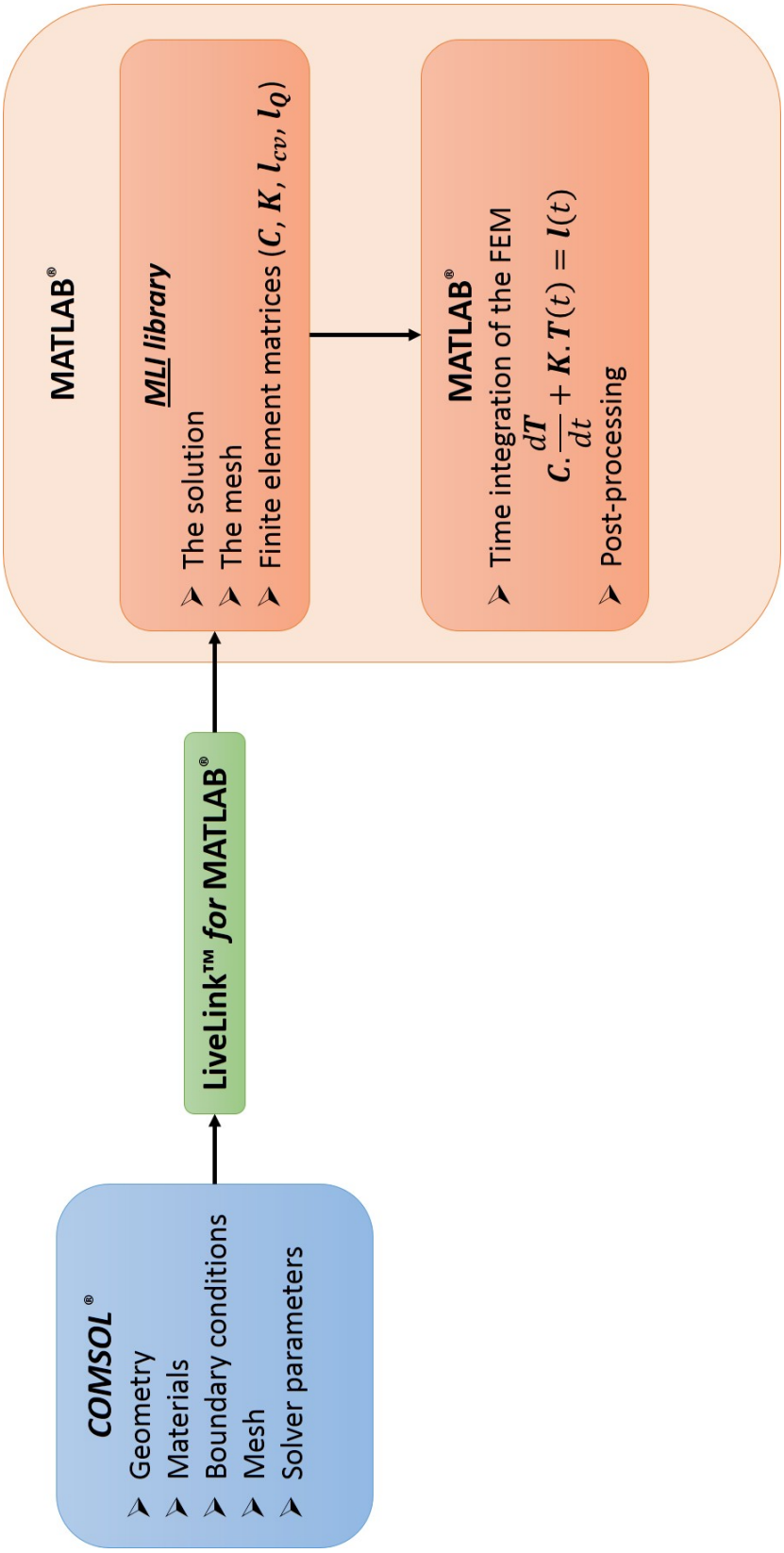


Figure 2.3 – Finite element model constructing using COMSOL and MATLAB

2.4 Conclusion

Governing equations describing the thermal behavior in an electronic package have been described as well as the coupling of heat transfer at fluid-solid boundaries. Space discretization using the finite element method of heat transfer equation in fluid and solid media have been described. The chosen time integration method is the Crank-Nicolson scheme. The finite element formulation of conjugate heat transfer is also illustrated for the simplified case in which the heat exchange at the fluid-solid interface is modeled by a convection coefficient. Finally, $COMSOL^{®}$ has been chosen to construct the finite element model. The main reason of this choice is the user ability to access to finite element matrices that are required to construct the reduced order model as it will be shown in the next chapter. The procedure of recovering the finite element matrices from $COMSOL^{®}$ to $MATLAB^{®}$ has been described briefly.

Bibliography

[1] Theodore L Bergman, Frank P Incropera, David P DeWitt, and Adrienne S Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[2] Donald D Gray and Aldo Giorgini. The validity of the boussinesq approximation for liquids and gases. *International Journal of Heat and Mass Transfer*, 19(5):545–551, 1976.

[3] Juan C Heinrich and Darrell W Pepper. The finite element method: basic concepts and applications. 2005.

[4] Alexandre Ern. *Aide-mémoire des éléments finis*. Dunod, 2005.

[5] Zhilin Li, Zhonghua Qiao, and Tao Tang. *Numerical Solution of Differential Equations: Introduction to Finite Difference and Finite Element Methods*. Cambridge University Press, 2017.

[6] Junuthula Narasimha Reddy and David K Gartling. *The finite element method in heat transfer and fluid dynamics*. CRC press, 2010.

[7] *LiveLink™ for MATLAB® User's Guide*.

Chapter 3

Model reduction methodology

The finite element model allows to compute the temperature field in a finite number of nodes belonging to a mesh. In order to improve the accuracy of the FEM it is recommended to use a fine mesh with large number of nodes. Therefore, computing the FEM becomes cumbersome.

In order to circumvent this limit, model reduction has emerged as a technic to lower the mathematical dimension of the high fidelity model (FEM) while preserving its accuracy.

In this chapter a non-exhaustive survey of reduction methods is carried out. The presented methods consist in changing the space (vector, frequency, ...) where the solution will be computed with a reduced cost. The two reduction methods that have been used in our study are described in details. In section 3.2, the direct reduced order model based on POD and Galerkin projection will be presented. In section 3.3, the inverse reduced order model based on POD and a minimization procedure will be depicted.

Sommaire

3.1 Review of model reduction methods

3.1.1 Introduction

When treating complex industrial systems, the field of interest (temperature, pressure, velocity or any other physical quantity) can not be determined analytically. For this reason numerical simulation or experimental tests are performed. In each case the field is determined in a finite number of locations that correspond to the nodes of the mesh or the measurement locations.

If we denote the field of interest by $\mathbf{u}(\mathbf{x}, t, \mathbf{p})$ then its decomposition in the standard (natural) basis can be written as:

$$\mathbf{u}(\mathbf{x}, t, \mathbf{p}) = \sum_{i=1}^{M} u_i(t)\ \mathbf{e}_i(\mathbf{x}) \tag{3.1}$$

where $\mathbf{e}_i(\mathbf{x}) = [0, 0, \ldots, 0, 1, 0, \ldots, 0] \in \mathbb{R}^M$ is the i-th column of the identity matrix $\mathbf{I}_M \in \mathbb{R}^{M\times M}$, $u_i(t)$ are the nodal values of $\mathbf{u}(\mathbf{x}, t, \mathbf{p})$ for $i = 1, \ldots, M$. $\mathbf{x}$ and M are respectively the coordinates and the number of nodes. $\mathbf{p}$ is a parameter vector that can correspond to boundary conditions, physical properties of the medium or numerical discretization parameters. In order to alleviate the notation, the symbol $\mathbf{p}$ will be omitted, but we have to keep in mind that the solution is computed for a given $\mathbf{p}$. Equation (3.1) describes the field of interest in the high dimensional space spanned by the basis $\{\mathbf{e}_i(\mathbf{x}_i)\}_{i=1,\ldots,M}$.

In classical numerical methods such as finite element method or finite volume method, the solution is computed in this high dimensional space. This task requires a large amount of computational resources, in particular when the physical domain is meshed finely.

The aim of model reduction techniques is to find another space of much lower dimension where the field $\mathbf{u}(\mathbf{x}, t)$ can be computed accurately. Consequently, model reduction consists in finding the solution as:

$$\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{r} a_i(t)\ \boldsymbol{\phi}_i(\mathbf{x}) \tag{3.2}$$

where $\mathbf{r << M}$. In equation (3.2), $\boldsymbol{\phi}_i(\mathbf{x}) \in \mathbb{R}^M$ are the components of the reduced basis and also called spatial modes. $a_i(t)$ are called modal amplitude or time amplitude corresponding to each mode. In most cases, we assume that a new solution can be projected on the precomputed spatial basis spanned by $\boldsymbol{\phi}_i(\mathbf{x})$. In order to determine a new solution (new parameter vector $\mathbf{p}$), its coordinates should be computed in the latter basis.

There are several reduction methods and the difference from one to another is attributed to the procedure used to determine the basis components $\boldsymbol{\phi}_i(\mathbf{x})$ and the temporal coordinates $a_i(t)$. A possible classification of the existing reducing methods is given as follows:

1. In the first class, the reduced order model is obtained by reducing the dimension of the detailed model. In our case, the detailed model is the space discretized heat equation by finite element method (called also state-space representation).

   The first step consists in constructing a reduced basis that can project accurately the field of interest. Many methods can be used to construct this basis. For instance POD allows to compute a basis from numerical or experimental results (empirical basis). The mathematical backgroud of this method will be given in sections 3.2.2. Such an approach does not

require the mass and stiffness matrices from the commercial software, therefore the POD can be considered as a non-intrusive method for deriving a basis. Other methods can be used such as the Krylov basis approach (section 3.1.6) and require to know the matrices of the finite element model. The latter information is not easy to extract from commercial softwares and these approaches can be described as intrusive methods.

In the second step, the detailed model (FEM) is projected on a basis, this operation is called Galerkin or Petrov projection. The Galerkin projection is described in section 3.2.3. In this class, the space discretization matrices (mass and stiffness matrices) should be known. This class of methods is described as intrusive.

2. In the second class, many numerical simulations are performed for a set of operating conditions. A common basis is extracted to represent the precomputed solutions. The modal amplitude in the common basis for each precomputed solution is stored. In order to determine a new solution, interpolation using radial basis functions or other methods is carried out on the already stored time amplitudes. This method has been shown efficient when dealing with steady-state solid mechanics problems [1–4]. A similar approach have been used to compute the Navier-Stokes equations in the case of a flow around a cylinder [5,6]. The class of methods is non intrusive because the reduced basis is computed from results and the modal amplitudes are obtained by interpolation from the stored ones. The main drawback of such approach is the need to perform many numerical simulations for several values of operating conditions which is time-consuming.

3. The third class of reduction methods consists in assuming that the reduced model has a mathematical form and trying to identify each term of the (non) linear system by means of a minimization algorithm. Modal identification method (presented in section 3.1.5), one of the most widely known method in heat transfer community, belongs to this class of reducing method. In this method, only reference results are needed to construct the reduced model, therefore this method can be described as non-intrusive. Reference results can be obtained by finite element simulations using commercial softwares or by experimental tests. This method will be explained in section 3.1.5.

4. In the fourth class, a parametric solution is computed using an enrichment algorithm. The solution is computed offline piece by piece for all possible operating conditions. In this method, no reference results or a priori numerical simulations are required to construct the solution. The proper generalized decomposition (PGD) belongs to this class of methods. In this method, the source code of the FEM should be modified, therefore this method is described as intrusive. More details about the PGD will be given in section 3.1.8.

In the above introduction, a global description of reduction methods is given. A possible classification has been made on model reduction methods. More details about the most known reduction methods will be exhibited in the following sections.

3.1.2 Proper Orthogonal Decomposition (POD)

The POD was originally developed by Pearson in 1901 [7] and is known in the literature under different names depending on the field of application [1].

In the field of statistical analysis, it is called principal component analysis (PCA). In the field of stochastic process modeling, it's called Karhunen Loève decomposition (KLD) and was developed around 1940 [1]. Jordan (in 1873) and Beltrami (in 1874) derived the same decomposition in a mathematical framework, and they called it singular value decomposition (SVD) [1].

POD is a powerful method to extract the dominant structure of high dimensional data. This allows to approximate the data on a low dimensional basis with high accuracy. The POD basis retains the regnant variations present in the data set. As an example for a two-dimensional data set, the two POD basis components are plotted on Figure 3.1:

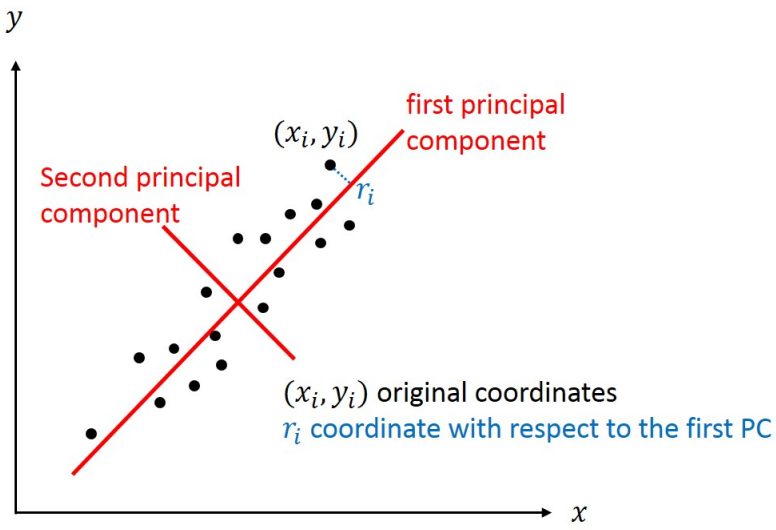


Figure 3.1 – POD in 2D

In this example, instead of representing the data by two components (x_i, y_i), it's possible to approximate the data by only one component r_i (we cut down the dimension from two to one). The POD in 2-dimensional space can be interpreted as a **rotation** of the original coordinate system. This concept can be extended to a set of high dimensional vectors.

Let us introduce the snapshots matrix **U**. This matrix contains the field $\mathbf{u}(t)$ at different instants t_i and for a given value of the parameter vector **p**, namely:

$$\mathbf{U}(\mathbf{p}) = [\mathbf{u}(t_1), \mathbf{u}(t_2), \ldots, \mathbf{u}(t_N)] = \begin{bmatrix} u(\mathbf{x}_1, t_1) & u(\mathbf{x}_1, t_2) & \ldots & u(\mathbf{x}_1, t_N) \\ u(\mathbf{x}_2, t_1) & u(\mathbf{x}_2, t_2) & \ldots & u(\mathbf{x}_2, t_N) \\ \vdots & \vdots & \ddots & \vdots \\ u(\mathbf{x}_M, t_1) & u(\mathbf{x}_M, t_2) & \ldots & u(\mathbf{x}_M, t_N) \end{bmatrix} \tag{3.3}$$

where M and N are respectively the number of nodes and time instants.

The aim of POD is to approximate the field $\mathbf{u}(t_i)$ in a vector space of dimension $k \leq M$, at any time t_i such that:

$$\begin{cases} \mathbf{u}(t_i) = \sum_{j=1}^{k} a_{ij}\, \boldsymbol{\phi}_j & (3.4a) \\ \sum_{i=1}^{N} \|\mathbf{u}(t_i) - \sum_{j=1}^{k} a_{ij}\boldsymbol{\phi}_j\|_2^2 \rightarrow minimum & (3.4b) \end{cases}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_i\}_{i=1,\dots,k}$ is the POD basis of order k. If $k = M$ then the POD basis is a complete basis and any field in $\mathbb{R}^M$ can be projected perfectly on this basis.

The coefficients a_{ij} are the coordinates of the field $\mathbf{u}(t_i)$ at each time instant t_i ($a_{ij} = a_j(t_i)$). The decomposition 3.4a can be seen as a separation of space and time variables as depicted in equation (3.2).

The minimization problem 3.4b is equivalent to solving the following eigenvalue problem [1]:

$$\mathbf{U}\mathbf{U}^T\, \boldsymbol{\phi}_i = \lambda_i\, \boldsymbol{\phi}_i \tag{3.5}$$

where λ_i is the eigenvalue of the mode number i. The baseline snapshots matrix $\mathbf{U}$ can be approximated using the first k modes, namely:

$$\mathbf{U} \approx \boldsymbol{\Phi}\mathbf{A} \;\; where \;\; \mathbf{A} = (a_{ij})_{1<i<N, 1<j<k}$$

In [1], it is shown that:

$$\sum_{i=1}^{N} \|\mathbf{u}(t_i) - \sum_{j=1}^{k} a_{ij}\, \boldsymbol{\phi}_j\|_2^2 = \sum_{i=1}^{N} \|\mathbf{u}(t_i)\|_2^2 - \sum_{i=1}^{k} \lambda_i \tag{3.6}$$

The left-hand side of equation (3.6) represents the Frobenius norm of the difference between snapshots matrix $\mathbf{U}$ and the matrix approximated by the first k POD modes. Equation (3.6) can be written in a different form:

$$\|\mathbf{U} - \boldsymbol{\Phi}\mathbf{A}\|_F^2 = \sum_{i=1}^{N} \|\mathbf{u}(t_i)\|_2^2 - \sum_{i=1}^{k} \lambda_i \tag{3.7}$$

Equation (3.7) shows that the eigenvalue has the same physical dimension as that of the square of $\mathbf{u}(t_i)$. In signal processing analysis the square of a signal is called energy. That's why λ_i will be called the energy of the mode i. In order to select the dominant modes, we calculate the energy contained in the mode i with respect to the total energy contained in the signal (the M modes). Therefore, the relative energy content (REC) is defined as:

$$REC(i) = \frac{\lambda_i}{\sum_{j=1}^{M} \lambda_j} \tag{3.8}$$

The number of dominant modes r required to keep 99% of the total energy is determined such that:

$$REC(r) \geq 0.99 \tag{3.9}$$

It is obvious from equation (3.7) that the quality of the POD approximation improves as the number of retained modes increases. In practice, eigenvalues decrease significantly, therefore few modes are required to represent accurately the data.

More details about this method will be presented in the section 3.2.2.

3.1.3 Singular Value Decomposition (SVD)

The singular value decomposition of a matrix $\mathbf{U}$ can be written as [8]:

$$\mathbf{U} = \mathbf{Z}\boldsymbol{\Sigma}\boldsymbol{\Psi}^T \tag{3.10}$$

where $\mathbf{Z}$ and $\boldsymbol{\Psi}$ are orthogonal matrices($\mathbf{Z}\mathbf{Z}^T = \mathbf{I}_M$ and $\boldsymbol{\Psi}\boldsymbol{\Psi}^T = \mathbf{I}_N$) having respectively the dimension $M \times M$ and $N \times N$. $\boldsymbol{\Sigma} \in \mathbb{R}^{M\times N}$ is a diagonal matrix containing the singular values which are arranged in decreasing order. Equation (3.10) can also be written (in the case $N < M$) as:

$$\mathbf{U} = \begin{pmatrix} z_{11} & \dots & z_{1M} \\ z_{21} & \dots & z_{2M} \\ z_{31} & \dots & z_{3M} \\ \vdots & \ddots & \vdots \\ z_{M1} & \dots & z_{MM} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \sigma_l & 0 & \dots & 0 \\ 0 & \dots & \dots & 0 & \sigma_{l+1} & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & \sigma_N \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \dots & \dots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix} \begin{pmatrix} \psi_{11} & \dots & \psi_{1N} \\ \psi_{21} & \dots & \psi_{2N} \\ \psi_{31} & \dots & \psi_{3N} \\ \vdots & \ddots & \vdots \\ \psi_{N1} & \dots & \psi_{NN} \end{pmatrix}^T \tag{3.11}$$

The rank of a matrix is the number of its non-zero singular values. Let's denote by l the rank of the matrix $\mathbf{U}$. This implies that $\sigma_{l+1} = \sigma_{l+2} = \dots = \sigma_M = 0$. Therefore equation (3.11) (the full form of the SVD) becomes :

$$\mathbf{U} = \begin{pmatrix} z_{11} & \dots & z_{1l} \\ z_{21} & \dots & z_{2l} \\ z_{31} & \dots & z_{3l} \\ \vdots & \ddots & \vdots \\ z_{M1} & \dots & z_{Ml} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_l \end{pmatrix} \begin{pmatrix} \psi_{11} & \dots & \psi_{1l} \\ \psi_{21} & \dots & \psi_{2l} \\ \psi_{31} & \dots & \psi_{3l} \\ \vdots & \ddots & \vdots \\ \psi_{N1} & \dots & \psi_{Nl} \end{pmatrix}^T \tag{3.12}$$

The SVD is similar to the separation of variables. In fact, the snapshots matrix depends on space and time. Each column vector of the matrix $\mathbf{Z}$ have the same dimension as the number of nodes and hence can be attributed to the space information in the snapshots matrix. Similarly, the matrix $\boldsymbol{\Psi}$ has the same dimension as the number of time instant as thus can be attributed to the temporal characteristic of the snapshots matrix.

The SVD and POD are equivalent. If we compute the product $\mathbf{U}\mathbf{U}^T$ using the expression of $\mathbf{U}$ given by equation (3.10), we obtain:

$$\mathbf{U}\mathbf{U}^T = \mathbf{Z}\boldsymbol{\Sigma}^2\mathbf{Z}^T$$

If we left multiply the last expression by $\mathbf{Z}$ and using the orthogonality of $\mathbf{Z}$ then we obtain :

$$\mathbf{U}\mathbf{U}^T\mathbf{Z} = \mathbf{Z}\mathbf{\Sigma}^2$$

The latter eigenvalue problem is identical to the POD eigenvalue problem given by equation (3.5). The eigenvalues of POD are the square of singular values and the $\mathbf{Z}$ matrix obtained by SVD is identical to the POD basis matrix $\mathbf{\Phi}$. Therefore, the singular value decomposition can be written as:

$$\mathbf{U} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T \tag{3.13}$$

The SVD decomposition can be viewed as a sum of modes :

$$\mathbf{U} = \sum_{i=1}^{rank(\mathbf{U})} \sigma_i \, \boldsymbol{\phi}_i \, \boldsymbol{\psi}_i^T$$

where σ_i is the i^{th} singular value, $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_i$ are the i^{th} column of matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, respectively. The term $\sigma_i \, \boldsymbol{\phi}_i \, \boldsymbol{\psi}_i^T$ is $M \times N$ matrix and represents the contribution of the i^{th} mode to the snapshots matrix. As the norm of $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_i$ is equal to one, the contribution of $\sigma_i \, \boldsymbol{\phi}_i \, \boldsymbol{\psi}_i^T$ is governed by the magnitude of the associated singular value σ_i. Accordingly, in order to select the dominant modes, analysis of the singular values distribution should be performed. More details about this method and the dominant modes selection criteria will be given in section 3.2.2.

3.1.4 Branch modes basis

Branch modes basis was originally used to investigate mechanical vibration of a system having many degrees of freedom [9]. This method was then used by Alain Neveu et al. [10] in order to reduce heat transfer problems.

Let us consider the heat equation and the boundary condition described as follows:

$$\begin{cases} c(\mathbf{x})\dfrac{\partial T(\mathbf{x},t)}{\partial t} = \nabla.(k(\mathbf{x})\nabla T(\mathbf{x},t)) \; \forall \mathbf{x} \in \mathbf{D} \\ k(\mathbf{x})\nabla(T(\mathbf{x},t)).\mathbf{n}(\mathbf{x}) = \varphi(\mathbf{x},t) + h(\mathbf{x})(T_f - T(\mathbf{x},t)) \; \forall \mathbf{x} \in \partial\mathbf{D} \end{cases}$$

where $\mathbf{D}$ is a physical domain of boundary $\partial\mathbf{D}$, $\mathbf{x}$ is a position vector belonging to $\mathbb{R}^3$ and $\mathbf{n}(\mathbf{x})$ is the normal to the boundary at the position $\mathbf{x}$. $c(\mathbf{x})$, $k(\mathbf{x})$ are respectively the volumetric heat capacity and the thermal conductivity of the medium. $\varphi(\mathbf{x},t)$ is an imposed heat flux on the boundary $\partial\mathbf{D}$. $h(\mathbf{x})$ and T_f are respectively the heat convection coefficient and the fluid temperature.

The branch eigenmodes problem applied to the thermal system is defined by the following equations [10]:

$$\begin{cases} \nabla.(k(\mathbf{x})\nabla V_i(\mathbf{x})) = \lambda_i \, c(\mathbf{x}) V_i(\mathbf{x}) \; \forall \mathbf{x} \in \mathbf{D} \\ -k(\mathbf{x})\nabla(V_i(\mathbf{x})).\mathbf{n}(\mathbf{x}) = \lambda_i \, \zeta V_i(\mathbf{x}) \; \forall \mathbf{x} \in \partial\mathbf{D} \end{cases}$$

where ζ is an arbitrary coefficient used in order to homogenize the physical dimension of the right term of the last equation. ζ is chosen constant such that $\displaystyle\int_{\partial\mathbf{D}} \zeta d\mathbf{x} = \int_{\mathbf{D}} c(\mathbf{x}) d\mathbf{x}$. ζ is called

Steklov parameter. Each eigenmode $V_i(\mathbf{x})$ is associated to its eigenvalue λ_i. We can notice that to obtain the Branch modes, we have to set $T(\mathbf{x},t)$ equal to $V_i(\mathbf{x})\ \exp(\lambda_i t)$ which means that the eigenvalues in this basis have the dimension of the inverse of time in contrast to the POD basis for which the eigenvalue have the dimension of the square of field of interest. The eigenmodes $V_i(x)$ form a complete basis for the space of functions that are continuous in $\mathbf{D} \cup \partial\mathbf{D}$ and are square-integrable functions. The eigenmodes $V_i(x)$ satisfy the othogonality relation:

$$< V_i, V_j >= \int_{\mathbf{D}} c(\mathbf{x}) V_i(\mathbf{x}) V_j(\mathbf{x}) d\mathbf{x} + \int_{\partial\mathbf{D}} \zeta V_i(\mathbf{x}) V_j(\mathbf{x}) d\mathbf{x} = \delta_{ij}$$

As the branch modes form a complete basis, any analytical solution (temperature) can be written as follows:

$$\forall \mathbf{x} \in \mathbf{D}\ \ T(\mathbf{x},t) = \sum_{i=1}^{\infty} a_i(t) V_i(\mathbf{x})$$

where $a_i(t)$ is the time amplitude related to the eigenmode $V_i(\mathbf{x})$. Otherwise, if the solution is obtained from spatial discretization, the vector $T(\mathbf{x},t)$ is composed of the temperature at the discretization nodes and has the dimension M. In such case, the temperature field $T(\mathbf{x},t)$ can be written as:

$$\forall \mathbf{x} \in \mathbf{D}\ \ T(\mathbf{x},t) = \sum_{i=1}^{M} a_i(t) V_i(\mathbf{x})$$

The aim of model reduction is to project the field T in a basis of lower dimension which require the selection of dominant modes. The process of selecting the dominant modes is described in [11]. We have to notice that this basis is independent of boundary conditions which have double effect: first the construction of the solution for given set of parameters will require a large number of modes compared to POD method. Secondly this basis will allow to simulate a wide range of configurations regardless of their input parameters.

For new operating conditions, the coordinates of a new solution $a_i(t)$ are determined by Galerkin projection of the finite element model on the retained branch modes. Details on Galerkin projection will be given in section 3.2.3.

This method has proven its robustness to analyze highly non-linear heat transfer problems [12]. In some industrial applications, the heat source can be difficult to estimate, for instance in brake systems [13] or cutting process [14]. Due to the high accuracy and the low computing time of the branch modes reduced order model, it has been combined with the inverse method in order to identify online the heat source magnitude [15, 16].

3.1.5 Modal Identification Method (MIM)

This method was first introduced by Daniel Petit et al. [17–19]. For the sake of simplicity an explanation of the MIM will be presented in the case of linear heat transfer. The modal identification method relies on two main stages:

1. **The first stage**

   This stage consists in defining a mathematical structure of the reduced model. In fact, when applying a space discretization (finite element, difference or volume methods) on the heat equation, a state space representation is obtained. The state space representation can be written as follows:

$$\begin{cases} \dot{\mathbf{T}}(t) = \mathbf{A}\,\mathbf{T}(t) + \mathbf{B}\,u(t) \\ \mathbf{Y}(t) = \mathbf{C}\,\mathbf{T}(t) \end{cases} \tag{3.14}$$

where $\mathbf{A}$ is the state matrix, $u(t)$ is the thermal input, $\mathbf{B}$ is the input vector, $\mathbf{C}$ is the observation matrix and $\mathbf{Y}(t)$ is the output vector.

Assuming that the matrix $\mathbf{A}$ is a diagonalizable matrix implies that:

$$\mathbf{A} = \mathbf{M}\,\mathbf{F}\,\mathbf{M}^{-1}$$

where $\mathbf{M}$ is the eigenvectors matrix of $\mathbf{A}$ and $\mathbf{F}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}$. A possible change of basis can be written as:

$$\mathbf{T}(t) = \mathbf{M}\,\mathbf{X}(t)$$

where $\mathbf{X}(t)$ are the coordinates of the temperature field in the modal basis $\mathbf{M}$. The aim of model reduction is to keep small the number of dominant modes (number of column of $\mathbf{M}$). This implies that the state representation of the reduced order model has a lower dimension and can be written as:

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}\,\mathbf{X}(t) + \mathbf{G}\,u(t) \\ \hat{\mathbf{Y}}(t) = \mathbf{H}\,\mathbf{X}(t) \end{cases}$$

where $\mathbf{F} = \mathbf{M}^{-1}\,\mathbf{A}\,\mathbf{M}$, $\mathbf{G} = \mathbf{M}^{-1}\,\mathbf{B}$ and $\mathbf{H} = \mathbf{C}\,\mathbf{M}$

2. **The second stage**

   This stage consists in identifying the parameters of the reduced model (matrices $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$) by minimizing a cost function $J_{red}(\mathbf{F}, \mathbf{G}, \mathbf{H})$. The cost function is defined as the difference between the output of the reduced model $\hat{\mathbf{Y}}(t)$ and the outputs of the reference model $\mathbf{Y}(t)$(detailed model or real process measurements), namely:

$$J_{red}(\mathbf{F}, \mathbf{G}, \mathbf{H}) = \sum_i \sum_j \left(Y_i(t_j) - \hat{Y}_i(t_j) \right)^2$$

   This stage depends on the efficiency of the minimization algorithm and its potential to identify the global minima. In [20] many minimization algorithms have been used and it is shown that the most efficient one is particle swarm optimization.

A scheme of the model identification method algorithm in the case of non-linear heat transfer is given in Figure 3.2 (The terms $\mathbf{\Psi}(\mathbf{T}(t))$ and $\mathbf{\Omega Z}(\mathbf{X}(t))$ correspond to the non-linearity of the models). In Figure 3.2, the use of sinusoids at each steady level of the input signal $U(t)$ aims to include the non-linear behavior in the identified reduced model.

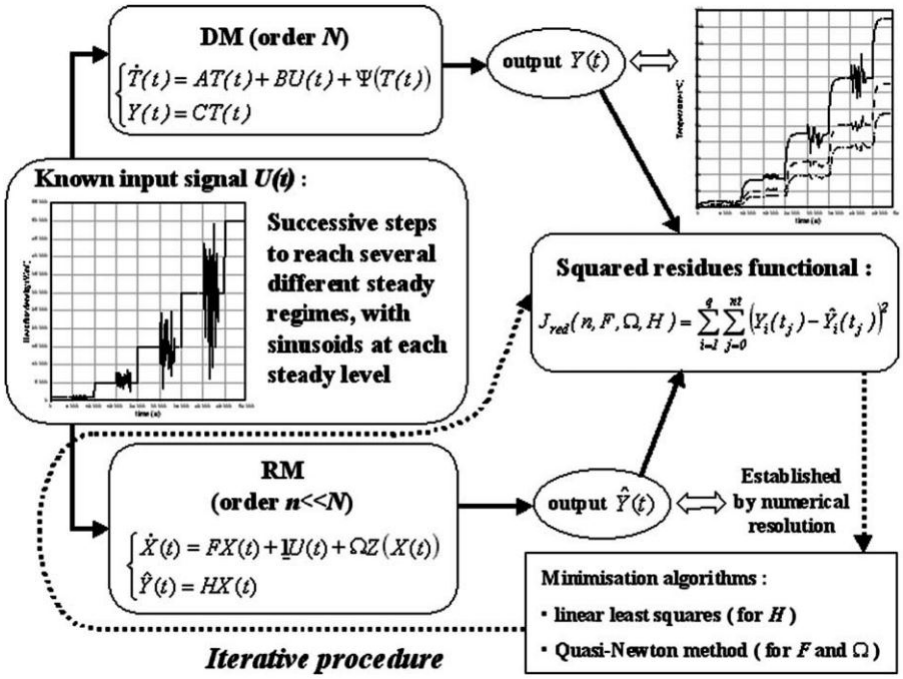


Figure 3.2 – Model Identification Method algorithm in the case of non-linear heat transfer [21]

We should bear in mind that no geometrical information and no physical properties appear explicitly in the MIM reduced model. This information is contained in the reference model data and therefore implicitly transmitted to the reduced model.

The MIM has been used to analyze many heat transfer problems. For instance, Manuel Girault and Daniel Petit [21,22] used the reduced order model based on MIM to solve an inverse non-linear heat conduction problem. This choice has made the inversion of heat conduction problem less time-consuming in order to identify the heat flux density. The MIM has been used to build reduced models for forced convection problems in the case heated flow over a backward-facing step [23]. In [23] a reduced order model for coupled fluid mechanic and heat transfer problem have been constructed using MIM. In [23] the CFD code Fluent® has been chosen to generate reference results in order to identify the reduced model parameters. The MIM does not require the source code of the industrial software, therefore it can be described as a non-intrusive reduction method.

The MIM has been compared to other reduction methods. In [24] E.Videcoq et al have compared the MIM to the branch eigenmodes reduction method in the case of 3D non-linear transient conduction problems. Numerical performances of the MIM (computing time and accuracy) was shown better than those of the branch eigenmodes reduced model. In [25], the MIM has been compared to the POD Galerkin method.

3.1.6 Krylov basis

This method is based on the approximation of the transfer function of the system on a certain range of frequency [26,27]. Applying the Laplace transform on equations (3.14) yields:

$$\mathbf{L}[\mathbf{Y}](s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{L}[u](s)$$

Where $\mathbf{L}[\,]$ and s represent respectively the Laplace operator and variable. Thus, the transfer function can be written as :

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

To pass from Laplace transformation to Fourier Transform we take $s = i2\pi f$ where f represents the frequency. Then, the transfer function $\mathbf{H}(s)$ can be approximated using Taylor expansion with respect to s. The Taylor expansion of the transfer function around s_0 is:

$$\mathbf{H}(s) = \sum_{k=0}^{\infty} \boldsymbol{\eta}_k(s_0)\frac{(s-s_0)^k}{k!}$$

where $\boldsymbol{\eta}_k(s_0)$ is called Markov moment around s_0 . For example:

- Around $s_0 = 0$: $\boldsymbol{\eta}_k(s_0) = \mathbf{C}\mathbf{A}^{-k-1}\mathbf{B}$. In this case the approximation is called Padé approximation. This approximation is suitable to compute the steady state response of the system (low frequency).
- $s_0 \in \mathbb{R}_{\setminus\{0,\infty\}}$: $\boldsymbol{\eta}_k(s_0) = \mathbf{C}(s_0\mathbf{I} - \mathbf{A})^{-k-1}\mathbf{B}$. In such case the approximation is called Shifted Padé approximation or rational interpolation and is aiming to approach the system response for a specific frequency band.
- $s_0 \to \infty$: $\boldsymbol{\eta}_k(s_0) = \mathbf{C}\mathbf{A}^k\mathbf{B}$. In this case, the approximation is called partial realization. These moments give a good approximation of high-frequency response of the system.

Model reduction consists in cutting down the previous sum at the order r, namely:

$$\mathbf{H}(s) \simeq \mathbf{H}_r(s) = \sum_{k=0}^{r} \boldsymbol{\eta}_k(s_0)\frac{(s-s_0)^k}{k!}$$

Markov moments allow to compute the transfer function in a low dimensional space. The transfer function links the output $\mathbf{Y}(t)$ and the input $u(t)$. In order to compute the state vector $\mathbf{T}(t)$, the matrix $\mathbf{C}$ should be omitted in the expression of Markov moments which yields the Krylov basis.

Krylov basis ($\mathcal{K}_r$) have different form depending on the operating frequency (s_0):

- In the case of Padé approximation:

$$\mathcal{K}_r(\mathbf{A}^{-1}, \mathbf{B}) = \{\mathbf{A}^{-1}\mathbf{B}, \mathbf{A}^{-2}\mathbf{B}, \cdots, \mathbf{A}^{-r}\mathbf{B}\}$$

- In the case of shifted Padé approximation:

$$\mathcal{K}_r((s_0\mathbf{I} - \mathbf{A})^{-1}, \mathbf{B}) = \{(s_0\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, (s_0\mathbf{I} - \mathbf{A})^{-2}\mathbf{B}, \cdots, (s_0\mathbf{I} - \mathbf{A})^{-r}\mathbf{B}\}$$

- In the case of partial realization:

$$\mathcal{K}_r(\mathbf{A}, \mathbf{B}) = \{\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \cdots, \mathbf{A}^{r-1}\mathbf{B}\}$$

Krylov vectors are non-orthogonal and in many cases linearly dependent. Thus, Arnoldi algorithm and Gram-Schmidt orthonormalization should be performed in order to extract a subset of Krylov basis.

3.1.7 Balanced realization basis

This method is used in the community of system control in order to approximate the linear dynamical system [27]. It permits to construct a reduced basis for local parameter values. The balanced realization basis is computed from the two Gramians of equations (3.14) defined as follows :

$$\mathbf{G}_c = \int_{t_0}^{t_f} \exp(\mathbf{A}.t)\mathbf{B}\mathbf{B}^T \exp(\mathbf{A}^T.t).dt$$

$$\mathbf{G}_o = \int_{t_0}^{t_f} \exp(\mathbf{A}^T.t)\mathbf{C}\mathbf{C}^T \exp(\mathbf{A}.t).dt$$

Where $\mathbf{G}_c \in \mathbb{R}^{M\times M}$ and $\mathbf{G}_o \in \mathbb{R}^{M\times M}$ are respectively the controllability and observability matrices.
The controllabilty and observability matrices are solutions of Lyapunov equations [27,28] :

$$\begin{cases} \dot{\mathbf{G}}_c = \mathbf{A}\mathbf{G}_c + \mathbf{G}_c\mathbf{A}^T + \mathbf{B}\mathbf{B}^T \\ -\dot{\mathbf{G}}_o = \mathbf{A}^T\mathbf{G}_o + \mathbf{G}_o\mathbf{A} + \mathbf{C}^T\mathbf{C} \end{cases}$$

When $t \to \infty$ the Gramians are defined if, and only if, the matrix $\mathbf{A}$ has eigenvalues with strictly negative real part. In such case the Gramians matrices verify the following modified Lyapunov equation:

$$\begin{cases} \mathbf{A}\mathbf{G}_c + \mathbf{G}_c\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = 0 \\ \mathbf{A}^T\mathbf{G}_o + \mathbf{G}_oA + \mathbf{C}^T\mathbf{C} = 0 \end{cases}$$

In practice, the Gramians matrices are calculated by solving Lyapunov equation. The formula of the Gramians with the integrals is used when the dimension of the linear system is too high.

The Balanced realization basis $\mathbf{J}$ is the basis in which:

- The Gramian of observability is equal to the Gramian of controllability: $\mathbf{J}\mathbf{G}_c\mathbf{J}^T = \mathbf{J}^{-T}\mathbf{G}_o\mathbf{J}^{-1}$
- The two gramians are diagonal: $\mathbf{J}\mathbf{G}_c\mathbf{J}^T = \mathbf{J}^{-T}\mathbf{G}_o\mathbf{J}^{-1} = \mathbf{\Gamma} = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_M \end{pmatrix}$

where σ_i are called Hankel singular values.

We can proove that:

$$\mathbf{J}\mathbf{G}_c\mathbf{G}_o\mathbf{J}^{-1} = \mathbf{\Gamma}^2$$

Which means that in order to calculate the Balanced realization basis we have to find the eigenvectors and eigenvalues of the matrix product of observability and controllabilty matrices $\mathbf{G}_c\mathbf{G}_o$.

The selection of dominant components of the balanced truncation basis is based on the following equations:

$$\begin{cases} \|\mathbf{H} - \mathbf{H}_r\|_\infty \leq 2\sum_{i=r+1}^{M} \sigma_i \\ \|\mathbf{Y}(t) - \mathbf{Y}_r(t)\|_2 \leq 2\sum_{i=r+1}^{M} \sigma_i \|u(t)\|_2 \end{cases}$$

The two previous equations show that the reduction error depends on the singular values of negelected modes as is the case in the POD or SVD. This error is quantified by the singular values of $\mathbf{G}_c\mathbf{G}_o$.

The drawback of this method is the computational cost of solving Lyapunov equations.

This method requires the matrices of the finite element model, which is not always easy, especially when using commercial softwares. Therefore this method can be described as intrusive reduction method.

3.1.8 Proper Generalized Decomposition (PGD)

This method can be found in the literature under two names [29] : the proper generalized decomposition (PGD) as introduced by Chinesta et al [30–32] or the LATIN method as introduced by Pierre Ladevèze et al [33].

The explanation of PGD will be given as follows. Let us denote by $u(x, y, z, t, \mathbf{p})$ the unknown field, then the PGD approximation consists in finding the solution having the following form (separated form):

$$u_m(x, y, z, t, \mathbf{p}) = \sum_{i=1}^{m} X_i(x)\, Y_i(y)\, Z_i(z)\, T_i(t)\, P_i^1(p_1) \cdots P_i^d(p_d) \tag{3.15}$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_d)$ is the parameter vector. $X_i(x)$, $Y_i(y)$, $Z_i(z)$, $T_i(t)$ and $P_i^j(p_j)$ are **a priori unknown** functions ($T_i(t)$ is not the temperature).

The decomposition 3.15 is similar to SVD (if the P_i^j functions are omitted).

The difference between SVD and PGD is that separated form of the solution for the SVD is obtained from the full order model solution whereas in PGD method these functions will be computed on the flight of the computational process. The PGD algorithm aims to compute each of the function $\{X_i(x), Y_i(y), Z_i(z), T_i(t), P_i^j(p_j)\}$ by an enrichment algorithm [34].

The first step is to compute the first mode from the boundary and initial conditions of the studied problem, namely :

$$u_1(x, y, z, t, \mathbf{p}) = X_1(x)\, Y_1(y)\, Z_1(z)\, T_1(t)\, P_1^1(p_1) \cdots P_1^d(p_d) \tag{3.16}$$

In equation (3.16), a single mode is used to model the boundary and initial conditions. To know the number of modes required for such purpose, reference [35] is useful. In [36] the authors used 3 modes to model the boundary conditions. A plot of the first 3 modes can be found in the same reference.

The kernel of this method is the algorithm that allows to enrich the solution; mainly how to find $u_{k+1}(x, y, z, t, \mathbf{p})$ knowing $u_k(x, y, z, t, \mathbf{p})$. The enrichment process is based on the alternating directions fixed point algorithm [32]. For a detailed explanation of this method the reader is referred to [30, 32, 37].

The PGD has been shown effective to solve many physical and industrial problems. In [36], the PGD have been combined with measurement devices in order to control a thermal process in real time. Such coupling between simulation and measuement is called Dynamic Data Driven Application System (DDDAS). In the same scope, the PGD has been used to identify in real time the heat diffusion parameter and reconstruct accurately the temperature field [38]. The PGD has been used to simulate the heat and mass transport in a square lid-driven cavity [39] and obtained results show good performances.

This method is highly intrusive because it requires all space discretization matrices (mass matrix, stiffness matrix, ...). Some studies focused on how to circumvent this limit and make the PGD suitable for industrial software [40].

3.1.9 Conclusion

Several reduction methods have been described in previous sections. The difference between these methods lies in the procedure used to construct the reduced basis on which the physical solution is projected. In the case of POD or SVD, the reduced basis is obtained from empirical results. Empirical results can be obtained from numerical simulation or experimental measurement. The main requirement to construct such basis, is to possess a set of data that represent the field of interest. That is why the POD basis is sometimes called empirical basis.

For the other methods (Branch modes basis, Krylov Basis, Balanced realization basis, PGD), the reduced basis is computed from the physical model (PDE or the state-space representation of the system). For these methods, it is required to know all the matrices of the state-space representation which is difficult to realize when using commercial numerical software.

The MIM is a reduction method consisting in identifying the model reduction parameters from empirical data. This method requires a set of data for the physical field of interest.

In our work, we choose to use the POD (or the SVD) for its convenience to deal with industrial applications. In fact, a commercial software has been used to compute the temperature field in the power module. Therefore, extraction of the POD basis is easy to perform from obtained results.

3.2 Proper Orthogonal Decomposition and Galerkin projection method

3.2.1 Introduction

In this section, a general description of the used reduction method is presented. The main steps of the ROM construction are presented on Figure 3.3.

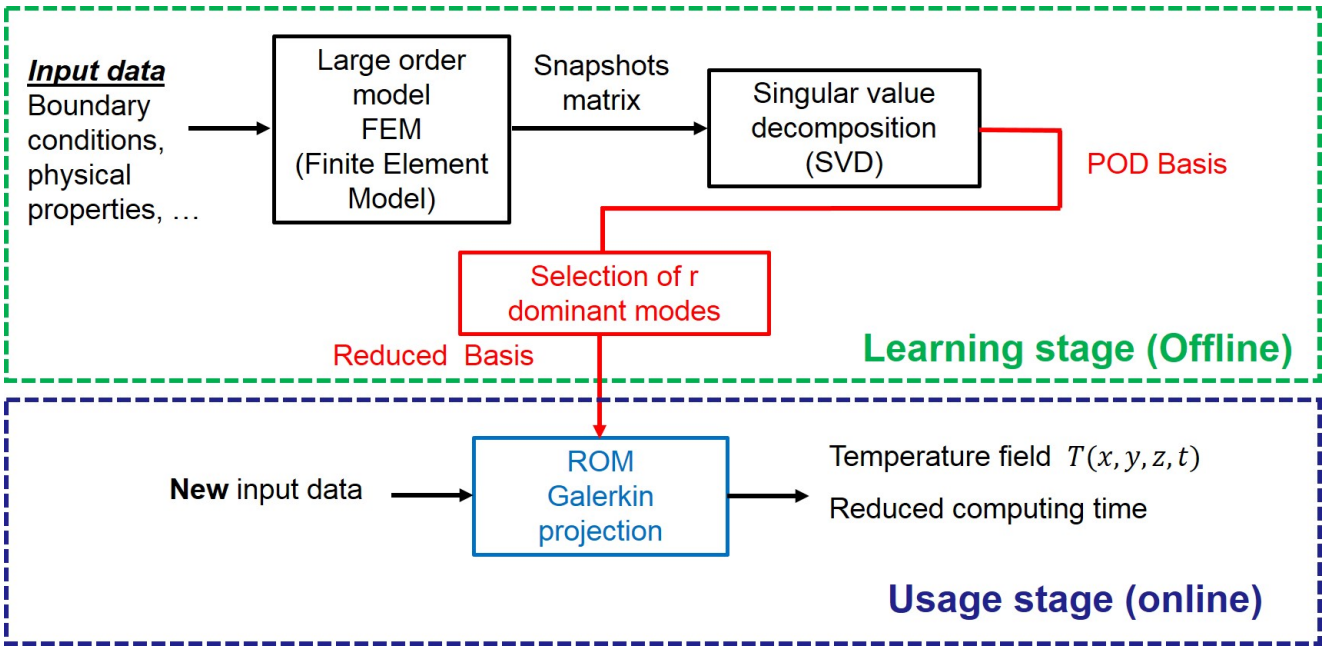


Figure 3.3 – Reduced order model scheme

The reduction method consists of two mains steps :

1. **The learning stage :**

   The aim of this stage is to construct a reduced basis that represents accurately the field of interest (in our case the temperature). The reduced basis stands for low dimensional basis. The first step of this stage is to run the finite element model for given operating conditions (boundary conditions, thermal load, ...). This step can be time-consuming because the finite element model has a dimension equal to the number of mesh nodes. The obtained results are stored in a snapshots matrix. Each column of this matrix contains the three-dimensional temperature field at given instants t_i. The singular value decomposition is then applied on the snapshots matrix. This decomposition represents each snapshot (the temperature field at t_i) as a linear combination of the POD modes. The number of POD modes is equal to the number of non-zero singular values of the SVD. The contribution of each POD mode is not the same. Therefore, a selection procedure should be carried out in order to keep the r dominant modes. This stage is done offline and only once because the SVD is time-consuming. The dominant modes (reduced basis) are the output of this stage and that will be used to construct the ROM.

2. **The usage stage :**

   In this stage, the solution will be computed in the reduced basis that has already been constructed during the learning stage. The finite element model is projected on the reduced basis (section 3.2.3). This (Galerkin) projection results in a state-space representation with a low dimension.

   Computing this latter is done online and is much less time-consuming than the FEM. The considered operating condition in this stage can be different from the ones used in the learning stage.

3.2.2 Derivation of the POD basis

In this section, the procedure used to derive the POD basis is detailed. Let's denote by $\mathbf{U}$ the snapshots matrix. Each column of the snapshots matrix corresponds to the temperature field $\mathbf{u}(t_i) \in \mathbb{R}^M$ at time t_i (called also snapshot at time t_i). In order to alleviate the notation, the snapshot at time t_i will be denoted by $\mathbf{u}^i$. The snapshots matrix can be written as:

$$\mathbf{U} = [\mathbf{u}^1, \mathbf{u}^2, ..., \mathbf{u}^N] \in \mathbb{R}^{M \times N}$$

where M is the number of nodes and N is the number of time instants.

The temperature field is denoted by $\mathbf{u}$ for the sake of generality. The same method can be applied to another type of physical field (velocity, pressure, energy, ...).

The aim of this method is to minimize the difference between the baseline solution (snapshots matrix) and the solution that we want to express as linear combination of the POD basis.

The solution expressed as a linear combination of the r first POD modes can be written :

$$\mathbf{u}_r^i = \sum_{j=1}^{r} a_{ij}\, \boldsymbol{\phi}_j \quad for \quad i \in \{1, 2, \ldots, N\}$$

where $\boldsymbol{\phi}_j$ are the POD (or spatial) modes and $a_{ij} = a_j(t_i)$ are the corresponding temporal amplitudes.

The error that this method aims to minimize is :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i - \sum_{j=1}^{r} a_{ij}\, \boldsymbol{\phi}_j\|_2^2 \tag{3.17}$$

The error (3.17) is the square of the Frobinus norm of the difference between the baseline snapshots matrix $\mathbf{U}$ and the snapshots matrix $\mathbf{U}_r$ containing $\mathbf{u}_r(t_i)$, namely :

$$E = \|\mathbf{U} - \mathbf{U}_r\|_F^2$$

Equations (3.17) can be written differently using the definition of the Euclidean norm :

$$E = \sum_{i=1}^{N} \langle \mathbf{u}^i - \sum_{j=1}^{r} a_{ij}\, \boldsymbol{\phi}_j, \mathbf{u}^i - \sum_{j=1}^{r} a_{ij}\, \boldsymbol{\phi}_j \rangle \tag{3.18}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. The Euclidean inner product is bilinear, therefore equation (3.18) becomes :

$$E = \sum_{i=1}^{N} \langle \mathbf{u}^i, \mathbf{u}^i \rangle - \sum_{i=1}^{N} \langle \mathbf{u}^i, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle - \sum_{i=1}^{N} \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \mathbf{u}^i \rangle + \sum_{i=1}^{N} \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle \tag{3.19}$$

Or the inner product is commutative, namely :

$$\langle \mathbf{u}^i, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle = \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \mathbf{u}^i \rangle$$

As a result, equation (3.19) becomes :

$$E = \sum_{i=1}^{N} \langle \mathbf{u}^i, \mathbf{u}^i \rangle - 2\ \sum_{i=1}^{N} \langle \mathbf{u}^i, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle + \sum_{i=1}^{N} \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle \tag{3.20}$$

The third term of equation (3.20) can expanded using the properties of the inner product (bilinear, distributive over vector addition) as follows :

$$\begin{aligned} \sum_{i=1}^{N} \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle &= \sum_{i=1}^{N} \sum_{j=1}^{r}\ a_{ij}\ \langle \boldsymbol{\phi}_j, \sum_{k=1}^{r} a_{ik}\ \boldsymbol{\phi}_k \rangle \\ &= \sum_{i=1}^{N} \sum_{j=1}^{r}\ a_{ij}\ \sum_{k=1}^{r} a_{ik}\ \langle \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \rangle \end{aligned} \tag{3.21}$$

The aimed basis is orthonormal, namely :

$$\langle \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \rangle = \delta_{jk}$$

where δ_{jk} is the kronecker symbol which is equal to 1 if $i = k$ and 0 if not. Therefore equation (3.21) becomes :

$$\begin{aligned} \sum_{i=1}^{N} \langle \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle &= \sum_{i=1}^{N} \sum_{j=1}^{r}\ a_{ij}\ \sum_{k=1}^{r} a_{ik}\ \delta_{jk} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{r} (a_{ij})^2 \end{aligned}$$

The error equation (3.20) becomes :

$$E = \sum_{i=1}^{N} \langle \mathbf{u}^i, \mathbf{u}^i \rangle - 2\ \sum_{i=1}^{N} \langle \mathbf{u}^i, \sum_{j=1}^{r} a_{ij}\ \boldsymbol{\phi}_j \rangle + \sum_{i=1}^{N} \sum_{j=1}^{r} (a_{ij})^2 \tag{3.22}$$

Equation (3.22) contains two unknowns $\boldsymbol{\phi}_j$ and a_{ij}. In order to minimize E, two derivatives with respect to $\boldsymbol{\phi}_j$ and a_{ij} should be performed.

Deriving equation (3.22) with respect to a_{lm} yields :

$$\begin{aligned}\frac{\partial E}{a_{lm}} &= -2\ \langle \mathbf{u}^l, \boldsymbol{\phi}_m \rangle + 2\ a_{lm} \\ &= -2\ \left(\mathbf{u}^l\right)^T\ \boldsymbol{\phi}_m + 2\ a_{lm}\end{aligned}$$

By setting $\frac{\partial E}{a_{lm}}$ equal to zero, we obtain :

$$a_{lm} = \left(\mathbf{u}^l\right)^T\ \boldsymbol{\phi}_m$$

Replacing the value of a_{ij} in equation (3.22), yields :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - 2\ \sum_{i=1}^{N} \langle \mathbf{u}^i, \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)\ \boldsymbol{\phi}_j \rangle + \sum_{i=1}^{N} \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)^2 \tag{3.23}$$

As $(\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)$ is a scalar and the inner product is bilinear, the second term of equation (3.23) can be expanded as follows :

$$\begin{aligned}\langle \mathbf{u}^i, \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)\ \boldsymbol{\phi}_j \rangle &= \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j) \langle \mathbf{u}^i, \boldsymbol{\phi}_j \rangle \\ &= \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)^2\end{aligned}$$

Therefore, equation (3.23) becomes :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{i=1}^{N} \sum_{j=1}^{r} (\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)^2 \tag{3.24}$$

By linear algebra [8] we can prove that :

$$(\left(\mathbf{u}^i\right)^T\ \boldsymbol{\phi}_j)^2 = \boldsymbol{\phi}_j^T\ \left(\mathbf{u}^i (\mathbf{u}^i)^T\right)\ \boldsymbol{\phi}_j$$

Therefore, equation (3.24) becomes :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{r}\ \boldsymbol{\phi}_j^T\ \left(\sum_{i=1}^{N} \mathbf{u}^i (\mathbf{u}^i)^T\right)\ \boldsymbol{\phi}_j \tag{3.25}$$

Let us introduce the covariance matrix :

$$\mathbf{C} = \sum_{i=1}^{N} \mathbf{u}^i (\mathbf{u}^i)^T = \mathbf{U}\,\mathbf{U}^T$$

Equation (3.25) becomes :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{r}\ \boldsymbol{\phi}_j^T\ \mathbf{C}\ \boldsymbol{\phi}_j \tag{3.26}$$

The minimization of E with respect to the temporal amplitude a_{ij} have been performed. In the following step, E will be minimized with respect to ϕ_i. An important constraint should be respected during the minimization procedure, which is the orthonormality of the POD basis. An optimization procedure with respect to constraint can be achieved using Lagrange multipliers method [41]. The quantity that we aim to maximize is :

$$\sum_{j=1}^{r} \boldsymbol{\phi}_j^T \, \mathbf{C} \, \boldsymbol{\phi}_j \tag{3.27}$$

The constraint that should be maintained is :

$$\|\boldsymbol{\phi}_j\|_2 = \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j = 1 \quad for \quad j = 1, \ldots, r \tag{3.28}$$

The Lagrange multiplier method [41–43] for the maximization of (3.27) with respect to the constraint (3.28) can be written :

$$\frac{\partial}{\partial \boldsymbol{\phi}_k} \left[\sum_{j=1}^{r} \boldsymbol{\phi}_j^t \, \mathbf{C} \, \boldsymbol{\phi}_j - \sum_{j=1}^{r} \lambda_j \, (\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j - 1) \right] = 0 \quad for \quad k = 1, \ldots, r \tag{3.29}$$

where λ_j are called Lagrange multipliers.

Equation (3.29) yields :

$$\mathbf{C} \, \boldsymbol{\phi}_k = \lambda_k \, \boldsymbol{\phi}_k \quad for \quad k = 1, \ldots, r \tag{3.30}$$

The matrix $\mathbf{C} \in \mathbb{R}^{M \times M}$ has M eigenvalues and vectors.

Equation (3.30) shows that the eigenvalues are identical to the Lagrange multipliers introduced in equation (3.29).

Eigenvectors verify equation (3.30), therefore the error given by equation (3.26) can be written as :

$$\begin{aligned} E &= \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{r} \boldsymbol{\phi}_j^T \, \left(\mathbf{C} \, \boldsymbol{\phi}_j \right) \\ &= \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{r} \boldsymbol{\phi}_j^T \, \lambda_j \, \boldsymbol{\phi}_j \end{aligned} \tag{3.31}$$

Or $\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j = 1$, therefore equation (3.31) becomes :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{r} \lambda_j \tag{3.32}$$

- **Remark 1: Round-off error in POD**

  If M POD modes are retained, the error E should be equal to zero. This is not the case, due to the round-off error of the inexact representation of numbers when using a computer.

This error will be denoted by E_{min} and is achieved when all the eigenvectors are retained $(r = M)$ for the approximation of snapshots. This error is given as follows :

$$E_{min} = \sum_{i=1}^{N} \|\mathbf{u}^i\|_2^2 - \sum_{j=1}^{M} \lambda_j \tag{3.33}$$

The roundoff error becomes important when the number of snapshots (N) increases.

- **Remark 2: Reconstruction error using r modes**

  The error resulting from using r modes is given by equation (3.32). If we subtract from this equation the roundoff error E_{min} given by equation (3.33), then we obtain :

$$E_r = E_{min} + \sum_{j=r+1}^{M} \lambda_j \tag{3.34}$$

  Equation (3.34) shows that the reduction error is governed by the eigenvalues of the neglected modes and the roundoff error.

  It can be noticed that no improvement in the snapshot approximation can be reached when the eigenvalues have the same order of magnitude as E_{min} because the computed modes are tainted by the roundoff error.

- **Remark 3: Reconstruction error bound**

  The error E given by equation (3.17) can be written as :

$$E = \sum_{i=1}^{N} \|\mathbf{u}^i - \mathbf{u}_r^i\|_2^2$$

  Or the maximum norm is bounded by the l^2-norm, namely :

$$\begin{aligned} \|\mathbf{u}^i - \mathbf{u}_r^i\|_\infty^2 &\leq \|\mathbf{u}^i - \mathbf{u}_r^i\|_2^2 \\ &\leq \sum_{i=1}^{N} \|\mathbf{u}^i - \mathbf{u}_r^i\|_2^2 \end{aligned}$$

  Therefore

$$\|\mathbf{u}^i - \mathbf{u}_r^i\|_\infty \leq \sqrt{E_r} \;\; for \;\; i = 1, \ldots, N \tag{3.35}$$

  When performing a singular value decomposition, the number of non-zero singular values is equal to the rank of $\mathbf{U}$. Therfore :

$$E_r = E_{min} + \sum_{j=r+1}^{rank(\mathbf{U})} \lambda_j$$

  Because $rank(\mathbf{U}) \leq min(M, N)$ and $\lambda_{r+1} \geq \lambda_{r+2} \geq \ldots \geq \lambda_{rank(\mathbf{U})}$, an error bound for E_r can be derived :

$$\begin{aligned} E_r \leq & \quad E_{min} + \sum_{j=r+1}^{min(M,N)} \lambda_j \\ \leq & \quad E_{min} + (min(M,N) - r)\ \lambda_{r+1} \end{aligned}$$

As a result, an error bound for the maximum error can be derived :

$$\|\mathbf{u}^i - \mathbf{u}_r^i\|_\infty \leq \sqrt{E_{min} + (min(M,N) - r)\ \lambda_{r+1}} \quad for \quad i = 1, \ldots, N \tag{3.36}$$

Equation (3.36), can be used to select the number of dominant modes that allow a maximum reconstruction error below or equal to $\sqrt{E_{min} + (min(M,N) - r)\ \lambda_{r+1}}$. When the round-off error is negligible, the reconstruction error is bounded by:

$$\|\mathbf{u}^i - \mathbf{u}_r^i\|_\infty \leq \sqrt{min(M,N) - r}\ \sigma_{r+1} \tag{3.37}$$

- **Remark 4: A different way to compute the POD basis when $N << M$**

  The matrix $\mathbf{C} = \mathbf{U}\,\mathbf{U}^T$ is a $M \times M$ matrix, where M is the number of nodes. Computing eigenvectors of this matrix is cumbersome in industrial applications where the number of nodes used to discretize the geometry is high. In order to overcome this limit, eigenvectors and eigenvalues of $\mathbf{U}^T\,\mathbf{U}$ should be computed (because $\mathbf{U}^T\,\mathbf{U} \in \mathbb{R}^{N \times N}$). The eigenvectors matrix and eigenvalues matrix are denoted respectively by $\mathbf{\Psi}$ and $\mathbf{\Lambda}$.

  It is shown in section 3.1.3 that the POD and SVD are equivalent and that the eigenvectors of $\mathbf{U}^T\,\mathbf{U}$ are the temporal modes of the SVD of $\mathbf{U}$. In addition, it is shown that eigenvalues of $\mathbf{U}^T\,\mathbf{U}$ are the square of the singular values of $\mathbf{U}$. Once $\mathbf{\Psi}$ and $\mathbf{\Lambda}$, are computed, the POD modes can be obtained by the following equation:

  $$\mathbf{\Phi} = \mathbf{U}\,\mathbf{\Psi}\,\mathbf{\Lambda}^{-1/2}$$

  Such procedure is feasable because $\mathbf{U}^T\,\mathbf{U} \in \mathbb{R}^{N \times N}$ where $M >> N$.

- **Remark 5: Orthogonal projection on the POD basis**

  Let us denote by $\mathbf{\Phi}_r$ the POD basis obtained for a given operating conditions (boundary conditions, FEM time step, ...) and by $\mathbf{U}_g$ the corresponding snapshots matrix.

  Then the snapshots matrix $\mathbf{U}_g$ can be decomposed as follows [44, 45] :

  $$\mathbf{U}_g = \mathbf{P}\,\mathbf{U}_g + (\mathbf{I} - \mathbf{P})\ \mathbf{U}_g \tag{3.38}$$

  where $\mathbf{P} = \mathbf{\Phi}_r\,\mathbf{\Phi}_r^T$ is an orthogonal projection on the subspace spanned by the first r POD modes $\mathbf{\Phi}_r$.

  Equation (3.38) can be written for each snapshot (at time t_i) as follows:

  $$\mathbf{u}(t_i) = \mathbf{P}\,\mathbf{u}(t_i) + (\mathbf{I} - \mathbf{P})\ \mathbf{u}(t_i) \tag{3.39}$$

  $\mathbf{P}\,\mathbf{u}(t_i)$ is the component belonging to the linear subspace spanned by the r first POD modes and $(\mathbf{I} - \mathbf{P})\ \mathbf{u}(t_i)$ belongs to the linear subspace spanned by the neglected POD modes.

This decomposition is illustrated on Figure 3.4.

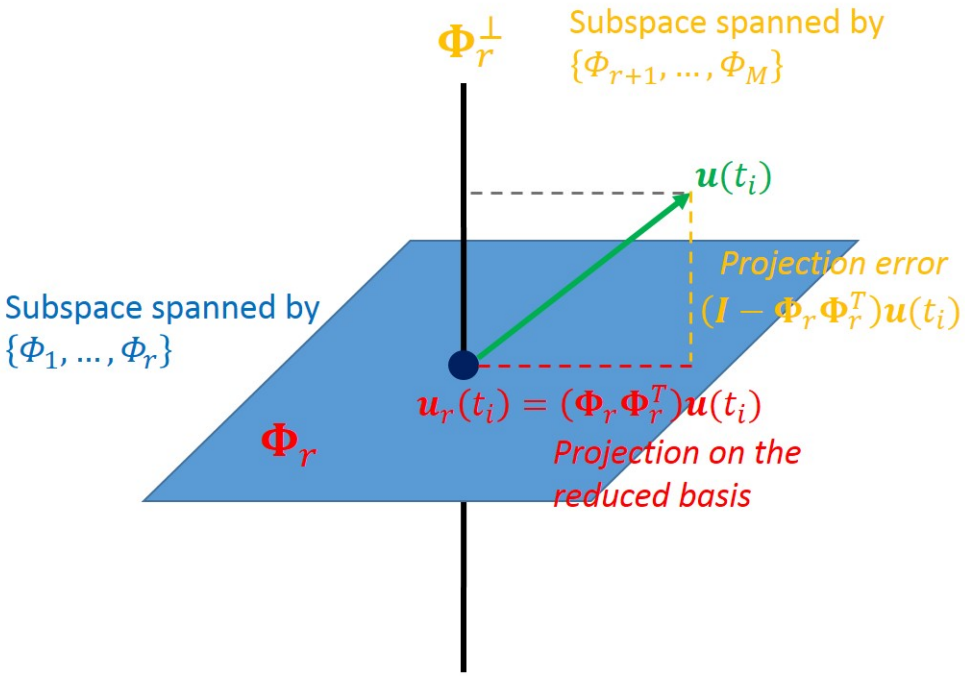


Figure 3.4 – Orthogonal decomposition of the snapshot $\mathbf{u}(t_i)$

The projection error of the snapshot $\mathbf{u}(t_i)$ on the r POD modes $\mathbf{\Phi}_r$ is given by :

$$\| (\mathbf{I} - \mathbf{P})\ \mathbf{u}(t_i)\|_2 \tag{3.40}$$

When the basis $\mathbf{\Phi}_r$ represents well the $\mathbf{U}_g$ data, this error is small.

If $\mathbf{\Phi}_r$ are the POD modes of $\mathbf{U}_g$ then the error given by equation (3.40) can be measured by the neglected eigenvalues as in equation (3.36).

- **Remark 6: Steady state snapshots matrix**

  The snapshots matrix can be constructed for the steady-state case. In such situation snapshots can correspond to the temperature field obtained for various boundary conditions $(\mathbf{p}_i)$, namely :

  $$\mathbf{U} = [\mathbf{u}(\mathbf{p}_1), \mathbf{u}(\mathbf{p}_2), ..., \mathbf{u}(\mathbf{p}_s)] \in \mathbb{R}^{M \times s}$$

  Such approach allows to reduce the steady-state finite element model. An application will be presented in Chapter 4.

- **Remark 7: POD basis interpolation**

  The POD basis is obtained for a given operating condition. Therefore the obtained basis represents well the snapshots used to construct the POD basis. In such situation, we may think of constructing many POD basis for a set of operating conditions and then make an interpolation of the reduced basis for a new operating condition. The issue with classical interpolation methods is that they yield a non-orthogonal basis. Therefore, D Amsallem

et al [46] proposed an interpolation in a tangent space to a Grassmann manifold . The Grassmann manifold is the set of POD basis computed for the set of operating conditions.

The concept of this method is presented on Figure 3.5 where **S** is a subspace spanned by a POD basis.

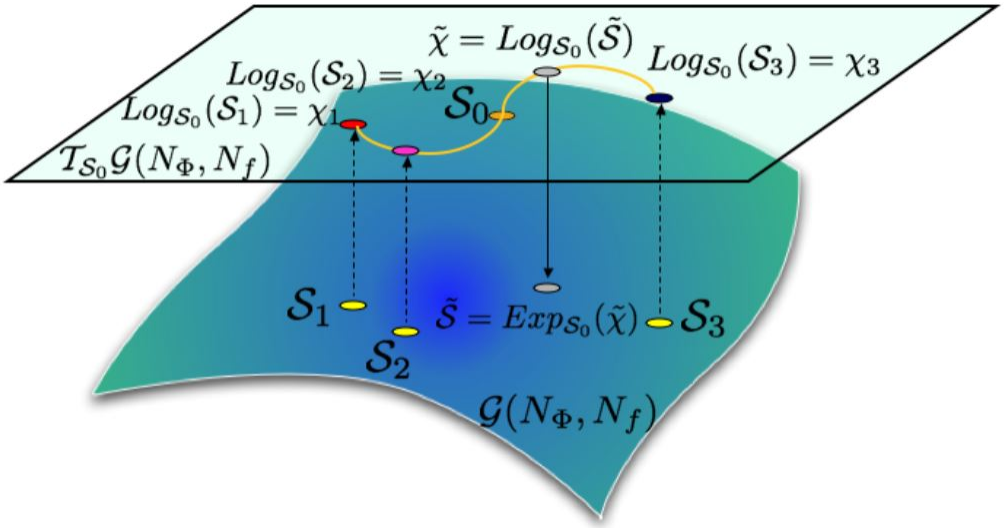


Figure 3.5 – Interpolation in the tangent plane [46]

The first step consists in transforming the POD basis from the Grassmann manifold (curved space) to the tangent space (straight space) by means of a logarithmic function. The tangent space is considered in a reference point. Then the interpolation can be performed in the tangent space. Thereafter, the interpolated POD basis can be obtained using an exponential transformation defined from the tangent space to the original space. This method has been used to interpolate the POD basis as function of the Mach number and the angle of attack for aeroelasticity applications [47].

Other interpolation methods of the POD basis exist. For instance Thuan Lieu et al. [48] used the angle between subspaces spanned by the set of precomputed POD basis in order to compute an adapted POD basis for the new operating condition.

3.2.3 Galerkin projection and the reduced order model

In this section, the ROM will be depicted for the two cases: steady state and transient state.

3.2.3.1 Steady state reduced order model

The steady state FEM is given by the following equation (Setting $\frac{d\mathbf{T}(t)}{dt} = 0$ in equation (2.24)):

$$(\mathbf{K}_0 + h\,\mathbf{K}_{cv})\;\mathbf{T} = P\,\mathbf{l}_Q + h\,T_{ext}\,\mathbf{l}_{cv} \tag{3.41}$$

where P, h and T_{ext} are respectively the overall power losses, the heat transfer coefficient and the external fluid temperature. The definition of matrices $\mathbf{K}_0$, $\mathbf{K}_{cv}$, $\mathbf{l}_Q$ and $\mathbf{l}_{cv}$ is given in Chapter 2. The aim of model reduction is to compute the temperature field in the reduced basis $\mathbf{\Phi}_r$, namely:

$$\mathbf{T} = \mathbf{\Phi}_r \, \mathbf{a}_r \tag{3.42}$$

where $\mathbf{a}_r \in \mathbb{R}^r$ is the coordinate vector of the temperature field $\mathbf{T}$ in $\mathbf{\Phi}_r$. We should keep in mind that $\mathbf{\Phi}_r$ is obtained from steady state snapshots for various operating conditions (P,h,T_{ext}). Inserting 3.42 in equation (3.41), yields :

$$(\mathbf{K}_0 + h \, \mathbf{K}_{cv}) \; \mathbf{\Phi}_r \, \mathbf{a}_r = P \, \mathbf{l}_Q + h \, T_{ext} \, \mathbf{l}_{cv} \tag{3.43}$$

Equation (3.43) consists of r unknowns ($\mathbf{a}_r$) and M equations. In order to remove this over-determination, equation (3.43) will be left-multiplied by $\mathbf{\Phi}_r^T$. This multiplication is called Galerkin projection. This yields the low dimensional linear system :

$$\mathbf{\Phi}_r^T \, (\mathbf{K}_0 + h \, \mathbf{K}_{cv}) \; \mathbf{\Phi}_r \, \mathbf{a}_r = P \, \mathbf{\Phi}_r^T \, \mathbf{l}_Q + h \, T_{ext} \, \mathbf{\Phi}_r^T \, \mathbf{l}_{cv} \tag{3.44}$$

Let us introduce the following matrices :

$$\begin{cases} \mathbf{K}_{0,r} = \mathbf{\Phi}_r^T \, \mathbf{K}_0 \, \mathbf{\Phi}_r \in \mathbb{R}^{r \times r} & \text{(3.45a)} \\ \mathbf{K}_{cv,r} = \mathbf{\Phi}_r^T \, \mathbf{K}_{cv} \, \mathbf{\Phi}_r \in \mathbb{R}^{r \times r} & \text{(3.45b)} \\ \mathbf{l}_{Q,r} = \mathbf{\Phi}_r^T \, \mathbf{l}_Q \in \mathbb{R}^r & \text{(3.45c)} \\ \mathbf{l}_{cv,r} = \mathbf{\Phi}_r^T \, \mathbf{l}_{cv} \in \mathbb{R}^r & \text{(3.45d)} \end{cases}$$

The reduced order model can be written as :

$$(\mathbf{K}_{0,r} + h \, \mathbf{K}_{cv,r}) \; \mathbf{a}_r = P \, \mathbf{l}_{Q,r} + h \, T_{ext} \, \mathbf{l}_{cv,r} \tag{3.46}$$

Equation (3.46) is a square linear system of order r. Matrices $\mathbf{K}_{0,r}$, $\mathbf{K}_{cv,r}$, $\mathbf{l}_{Q,r}$ and $\mathbf{l}_{cv,r}$ are computed only once. The overall stiffness matrix $(\mathbf{K}_{0,r} + h \, \mathbf{K}_{cv,r})$, and the overall load vector $(P \, \mathbf{l}_{Q,r} + h \, T_{ext} \, \mathbf{l}_{cv,r})$ are computed for any new value of operating conditions (P, h, T_{ext}). Solving (3.46) is not time-consuming because $r << M$. Once the coordinates vector $\mathbf{a}_r$ is computed, the temperature field can be obtained by equation (3.42).

3.2.3.2 Transient reduced order model

The finite element model (State-space representation) resulting from the space discretization of the transient heat equation and given by equation (2.24) can be written as follows :

$$\mathbf{C} \, \frac{d\mathbf{T}(t)}{dt} + (\mathbf{K}_0 + h \, \mathbf{K}_{cv}) \; \mathbf{T}(t) = P(t) \, \mathbf{l}_Q + h \, T_{ext} \, \mathbf{l}_{cv} \tag{3.47}$$

In order to solve equation (3.47), an initial condition is needed, that will be denoted by :

$$\mathbf{T}(t = 0) = \mathbf{T}_0 \tag{3.48}$$

The expressions of matrices $\mathbf{C}$, $\mathbf{K}_0$, $\mathbf{K}_{cv}$, $\mathbf{l}_Q$ and $\mathbf{l}_{cv}$ are given in Chapter 2.

The model reduction method consists in searching the solution in the low dimensional basis, namely :

$$\mathbf{T}(t) = \mathbf{\Phi}_r \, \mathbf{a}_r(t) \tag{3.49}$$

This can be seen as a basis change from the standard basis of $\mathbb{R}^M$ to the basis spanned by the POD vectors. $\mathbf{a}_r(t)$ is the coordinates vector of the temperature $\mathbf{T}(t)$ in the reduced basis. These coordinates depend on time. $\mathbf{\Phi}_r$ is the reduced POD basis obtained from transient snapshots.

Inserting equation (3.49) in (3.47) yields :

$$\mathbf{C}\,\frac{d\,\mathbf{\Phi}_r\,\mathbf{a}_r(t)}{dt} + (\mathbf{K}_0 + h\,\mathbf{K}_{cv})\;\mathbf{\Phi}_r\,\mathbf{a}_r(t) = P(t)\,\mathbf{l}_Q + h\,T_{ext}\,\mathbf{l}_{cv} \tag{3.50}$$

As the POD basis depends only on space, equation (3.50) becomes :

$$\mathbf{C}\,\mathbf{\Phi}_r\,\frac{d\,\mathbf{a}_r(t)}{dt} + (\mathbf{K}_0 + h\,\mathbf{K}_{cv})\;\mathbf{\Phi}_r\,\mathbf{a}_r(t) = P(t)\,\mathbf{l}_Q + h\,T_{ext}\,\mathbf{l}_{cv} \tag{3.51}$$

The system (3.51) is over-determined as it contains M equations and r unknowns where $r < M$. In order to remove this over-determination, we left-multiply the last equation by the matrix $\mathbf{W}^T \in \mathbb{R}^{r\times M}$. The obtained equation can be written as follows :

$$\mathbf{W}^T\,\mathbf{C}\,\mathbf{\Phi}_r\,\frac{d\,\mathbf{a}_r(t)}{dt} + \left(\mathbf{W}^T\,\mathbf{K}_0 + h\,\mathbf{W}^T\,\mathbf{K}_{cv}\right)\;\mathbf{\Phi}_r\,\mathbf{a}_r(t) = P(t)\,\mathbf{W}^T\,\mathbf{l}_Q + h\,T_{ext}\,\mathbf{W}^T\,\mathbf{l}_{cv} \tag{3.52}$$

If $\mathbf{W} = \mathbf{\Phi}_r$ the projection is called Galerkin projection, if also $\mathbf{\Phi}_r^T\mathbf{\Phi}_r = \mathbf{I}$ the vectors of $\mathbf{\Phi}_r$ are orthonormal and the projection is called orthonormal Galerkin projection.

If $\mathbf{W} \neq \mathbf{\Phi}_r$ the projection is called Petrov-Galerkin projection. If $\mathbf{W}^T\mathbf{\Phi}_r = \mathbf{I}$ vectors of $\mathbf{W}$ and $\mathbf{\Phi}$ are bi-orthonormal and the projection is called bi-orthonormal Petrov Galerkin projection. For instance, in [49] Petrov Galerkin projection has been used to reduce a fluid dynamics finite element model.

In our case $\mathbf{W} = \mathbf{\Phi}_r$, therefore equation (3.52) becomes :

$$\mathbf{\Phi}_r^T\,\mathbf{C}\,\mathbf{\Phi}_r\,\frac{d\,\mathbf{a}_r(t)}{dt} + \left(\mathbf{\Phi}_r^T\,\mathbf{K}_0\,\mathbf{\Phi}_r + h\,\mathbf{\Phi}_r^T\,\mathbf{K}_{cv}\,\mathbf{\Phi}_r\right)\,\mathbf{a}_r(t) = P(t)\,\mathbf{\Phi}_r^T\,\mathbf{l}_Q + h\,T_{ext}\,\mathbf{\Phi}_r^T\,\mathbf{l}_{cv} \tag{3.53}$$

Equation (3.53) can be written in a compact form :

$$\mathbf{C}_r\,\frac{d\,\mathbf{a}_r(t)}{dt} + (\mathbf{K}_{0,r} + h\,\mathbf{K}_{cv,r}\,)\;\mathbf{a}_r(t) = P(t)\,\mathbf{l}_{Q,r} + h\,T_{ext}\,\mathbf{l}_{cv,r} \tag{3.54}$$

where

$$\begin{cases} \mathbf{C}_r = \mathbf{\Phi}_r^T\,\mathbf{C}\,\mathbf{\Phi}_r \;\in \mathbb{R}^{r\times r} & \text{(3.55a)}\\ \mathbf{K}_{0,r} = \mathbf{\Phi}_r^T\,\mathbf{K}_0\,\mathbf{\Phi}_r \;\in \mathbb{R}^{r\times r} & \text{(3.55b)}\\ \mathbf{K}_{cv,r} = \mathbf{\Phi}_r^T\,\mathbf{K}_{cv}\,\mathbf{\Phi}_r \;\in \mathbb{R}^{r\times r} & \text{(3.55c)}\\ \mathbf{l}_{Q,r} = \mathbf{\Phi}_r^T\,\mathbf{l}_Q \;\in \mathbb{R}^{r} & \text{(3.55d)}\\ \mathbf{l}_{cv,r} = \mathbf{\Phi}_r^T\,\mathbf{l}_{cv} \;\in \mathbb{R}^{r} & \text{(3.55e)} \end{cases}$$

Therefore, model reduction brings down the dimension of the ordinary differential equations from M to r where $r << M$.

In practice, matrices (3.55) are computed once. For new operating conditions ($P(t)$,h,T_{ext}), the overall reduced stiffness matrix and reduced load vector should be computed, as follows :

$$\begin{cases} \mathbf{K}_r(h) = \mathbf{K}_{0,r} + h\,\mathbf{K}_{cv,r} & (3.56a) \\ \mathbf{l}_r(P(t), h, T_{ext}) = P(t)\,\mathbf{l}_{Q,r} + h\,T_{ext}\,\mathbf{l}_{cv,r} & (3.56b) \end{cases}$$

By solving the dynamical system given by equation (3.54) the coordinate vector $\mathbf{a}_r(t)$ is obtained. Then the temperature field is obtained using equation (3.49).

The initial condition is expressed as a function of $\mathbf{T}$. An initial condition on the coordinate vector $\mathbf{a}_r(t)$ should be derived in order to integrate numerically equation (3.54).

Taking equation (3.49) at $t = 0$ and left-multiplying it by $\mathbf{\Phi}_r^T$ ($\mathbf{\Phi}_r^T\,\mathbf{\Phi}_r = \mathbf{I}_r$) yields an initial condition on the coordinates vector than can be written as follows:

$$\mathbf{a}_r(t=0) = \mathbf{\Phi}_r^T\,\mathbf{T}(t=0) \tag{3.57}$$

In conclusion, the reduced order model is described by equations (3.58) :

$$\begin{cases} \mathbf{C}_r\,\dfrac{d\,\mathbf{a}_r(t)}{dt} + \mathbf{K}_r(h)\,\mathbf{a}_r(t) = \mathbf{l}_r(P(t), h, T_{ext}) & (3.58a) \\ \mathbf{a}_r(t=0) = \mathbf{\Phi}_r^T\,\mathbf{T}_0 & (3.58b) \\ \mathbf{T}(t) = \mathbf{\Phi}_r\,\mathbf{a}_r(t) & (3.58c) \end{cases}$$

- **Remark**

  The reduced order model consists of r equations. This will require $4r^2$ operations at each time step when using the Crank-Nicolson numerical scheme. In order to obtain the physical solution, the matrix vector product $\mathbf{T}(t) = \mathbf{\Phi}_r\,\mathbf{a}_r(t)$ should be computed which corresponds to $M(2r-1)$ operations. Therefore, the entire number of operations is $4r^2 + M(2r-1)$. We recall that the time integration of the finite element model using the Crank-Nicolson scheme requires $4M^2$ operations.

3.3 Inverse reduced order model (IROM)

3.3.1 Introduction

In the previous section, the reduced order model is **direct** because it requires as input informations the boundary conditions, the heat source and the initial condition.

In many practical situations, boundary conditions may be inavailable or incorrectly estimated. To overcome this limitation, the inverse reduced order model (IROM) will be introduced. The main feature of the IROM is its capability to establish a relationship between the known (available) and the unknown (field of interest that we aim to compute) information.

The method to be described is called here Inverse Reduced Order Model (IROM) but in the literature it is known as gappy proper orthogonal decomposition [50] or Karhunen Loeve procedure for gappy data [51]. The term "inverse" stands for the fact that in our application the measured temperature allows to compute the unknown information (temperature in inaccessible locations).

This section is focusing on how to establish a direct and simple mathematical relationship between the known information (measured temperature) and the unknown information (missed temperature). The model is based on two main stages: the learning stage based on POD and the usage stage based on a minimization algorithm. The block diagram describing the IROM is given in Figure 3.6.

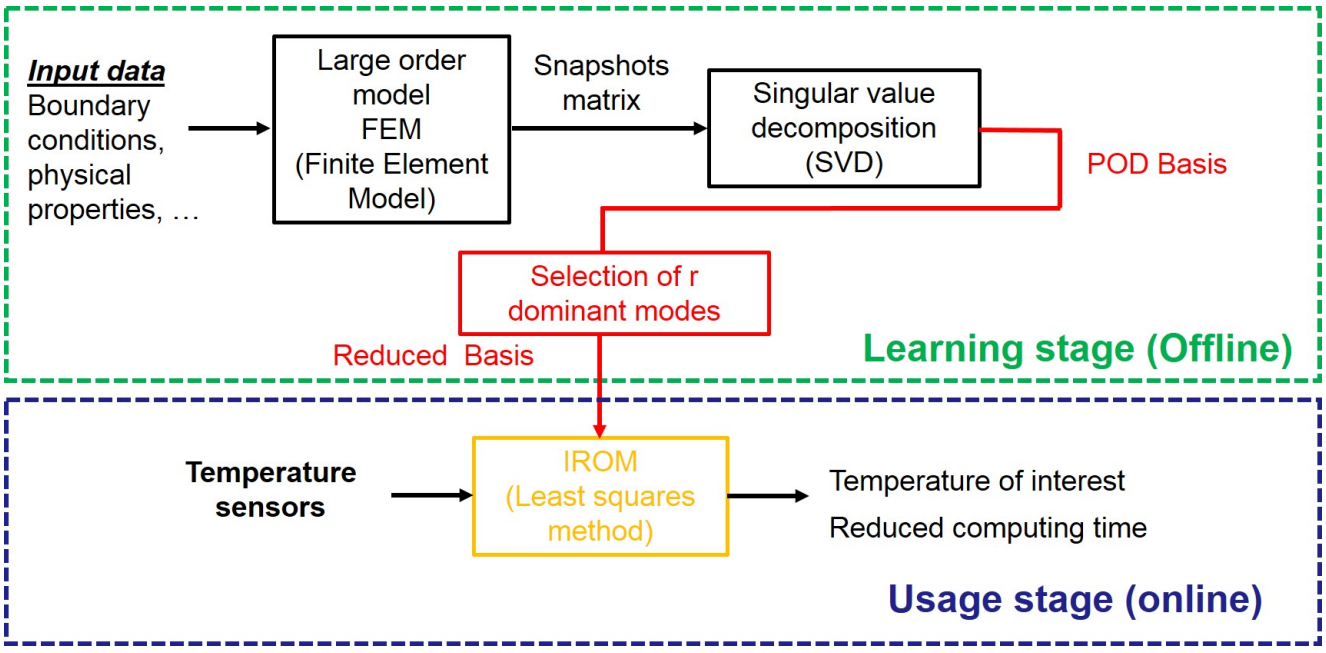


Figure 3.6 – Block diagram of the Inverse Reduced Order Model

The description of each stage will be given as follows :

- **The learning stage :**

  During this stage, a simulation with a reference model (FEM) is performed for given operating conditions. Obtained results include all nodes (locations). In this step, we outline the fact that the information is computed everywhere. The obtained results are stored in the snapshots matrix. The singular value decomposition is then applied to the snapshots

matrix. This decomposition represents each snapshot (the temperature field at t_i) as a linear combination of the POD modes. The contribution of each POD basis component is not the same. Therefore, a selection procedure should be carried out in order to keep the r dominant modes. This stage is done offline and only once. The dominant modes (reduced basis) are the output of this stage and that will be used to construct the IROM.

- **The usage stage :**

  In this stage, only the information (temperature) in a subset of mesh nodes are known. The aim of this stage is to compute the missing temperature (aimed by the user) from the measured temperature (obtained from sensors). The reduced basis establishes a coupling between the known part and the unknown part. The usage stage takes advantage of this coupling and derives a simple mathematical relation between the knowns and the unknowns. This relation is based on the fact that known and unknown part have the same coordinates in the reduced basis. Further mathematical details will be given in section 3.3.3.

This method have been used in many field:

- Improve the quality of marred data by filling its gaps in a reasonable manner [51].
- Reconstruct the entire pressure field from the airfoil surface pressure distribution [50, 52, 53].
- Enhance the gappy particle image velocimetry (PIV) data of shallow rectangular cavity flow at different Mach number [54].

In Chapter 5, an application of the IROM will be given in the case of the three-legs power module. MOSFETs temperature will be computed from temperature sensors.

3.3.2 The learning stage (POD)

The learning stage for the IROM is similar to the learning stage of the ROM based on Galerkin projection. For more details about the construction of the reduced basis using POD, the reader can refer to section 3.2.2.

The temperature field is computed on the finite element mesh. We can choose to work on the entire mesh or select locations of interest for instance the maximum temperature of each MOSFET and the mean temperature of each sensor. The SVD is applied on the snapshots matrix which yields the POD modes. After selecting the dominant POD modes, the reduced basis will be denoted by $\mathbf{\Phi}_r$.

The temperature field can be written as:

$$\mathbf{u}_r(t_i) = \sum_{j=1}^{r} a_j(t_i)\, \boldsymbol{\phi}_j = \mathbf{\Phi}_r\, \mathbf{a}_r(t_i) \tag{3.59}$$

The temperature field is denoted by $\mathbf{u}_r$ for the sake of generality because this method can be applied to another physical field (pressure, density, velocity, energy, ...). $\mathbf{u}_r \in \mathbb{R}^M$ where M can be the number of nodes or the number of locations of interest (sensors and nodes where the temperature is sought).

3.3.3 The usage stage

The field $\mathbf{u}_r$ mentioned in the previous section contains the field of interest everywhere. In the usage stage, only a part of the vector $\mathbf{u}_r$ is known, for instance the temperature in sensors locations. Equation (3.59) shows that all elements of $\mathbf{u}_r$ have the same coordinates in the reduced basis $\mathbf{\Phi}_r$. The idea of the IROM consists in extracting these coordinates from the known part of $\mathbf{u}_r$. In this section, the procedure to compute these coordinates will be depicted.

The first step in this stage is to construct matrices allowing to retrieve the known part and the unknown part of the overall temperature vector $\mathbf{u}_r$. Let us denote by $\mathcal{S}_{kn} \in \mathbb{R}^{M_{kn} \times M}$ the selection matrix of the known part and by $\mathcal{S}_{unkn} \in \mathbb{R}^{M_{unkn} \times M}$, where M_{kn} and M_{unkn} are the number of nodes where the temperature is known and unknown respectively ($M = M_{kn} + M_{unkn}$). The known part $\mathbf{u}_{r,kn}(t_i)$ and the unknown part $\mathbf{u}_{r,unkn}(t_i)$ of the temperature field are given by:

$$\mathbf{u}_{r,kn}(t_i) = \mathcal{S}_{kn}\, \mathbf{u}_r(t_i) \tag{3.60}$$

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}\, \mathbf{u}_r(t_i) \tag{3.61}$$

Let us denote by $(\alpha_1, \alpha_2, \ldots, \alpha_{M_{kn}})$ the set of indices of the nodes where the temperature is known and $(\beta_1, \beta_2, \ldots, \beta_{M_{unkn}})$ the set of indices of the nodes where the temperature is unknown. The selection matrix $\mathcal{S}_{kn}$ is given by equation (3.62):

$$\mathcal{S}_{kn} = \left[\mathbf{e}_{\alpha_1}, \mathbf{e}_{\alpha_2}, \ldots, \mathbf{e}_{\alpha_{M_{kn}}}\right]^T \tag{3.62}$$

where $\mathbf{e}_{\alpha_i} = [0, \ldots, 0, 1, 0, \ldots, 0] \in \boldsymbol{R}^M$ is the α_i-th column of the identity matrix $\mathbf{I}_M \in \boldsymbol{R}^{M \times M}$.

The selection matrix $\mathcal{S}_{unkn}$ is given by equation (3.63):

$$\mathcal{S}_{unkn} = \left[\mathbf{e}_{\beta_1}, \mathbf{e}_{\beta_2}, \ldots, \mathbf{e}_{\beta_{M_{unkn}}}\right]^T \tag{3.63}$$

where $\mathbf{e}_{\beta_i} = [0, \ldots, 0, 1, 0, \ldots, 0] \in \boldsymbol{R}^M$ is the β_i-th column of the identity matrix $\mathbf{I}_M \in \boldsymbol{R}^{M \times M}$.

The entire temperature field can be written in the reduced basis as:

$$\mathbf{u}_r(t_i) = \sum_{j=1}^{r} a_j(t_i)\, \boldsymbol{\phi}_j = \mathbf{\Phi}_r\, \mathbf{a}_r(t_i) \tag{3.64}$$

Equation (3.64) shows that the known part and the unknown part have the same coordinates in the reduced basis $\mathbf{\Phi}_r$. This fact is the main idea behind the IROM construction. The coordinates $\mathbf{a}_r(t_i)$ of the entire solution will be computed from the known part $\mathbf{u}_{r,kn}(t_i)$. Inserting equation (3.64) in (3.60) and (3.61) yields the two following equations:

$$\mathbf{u}_{r,kn}(t_i) = \mathcal{S}_{kn}\, \mathbf{\Phi}_r\, \mathbf{a}_r(t_i) \tag{3.65}$$

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}\, \mathbf{\Phi}_r\, \mathbf{a}_r(t_i) \tag{3.66}$$

The known temperature provided online by temperature sensors will be denoted by $\mathbf{u}_{r,kn}^{online}(t_i)$. The only unknown in equation (3.65) is the coordinates vector $\mathbf{a}_r(t_i)$. In order to compute it, the following minimization problem is solved:

$$\|\mathbf{u}_{r,kn}^{online}(t_i) - \mathcal{S}_{kn}\, \mathbf{\Phi}_r\, \mathbf{a}_r(t_i)\|_2 \rightarrow min \tag{3.67}$$

The problem (3.67) consists of M_{kn} equations and r unknowns, which is not a square linear system of equations. Therefore, a minimization algorithm should be carried out in order to solve (3.67). A possible way is to compute the pseudo inverse of $\mathcal{S}_{kn}\,\mathbf{\Phi}_r$ that will be denoted by $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^{\dagger}$. Therefore, the coordinates of the new solution in the reduced basis are :

$$\hat{\mathbf{a}}_r(t_i) = (\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^{\dagger}\ \mathbf{u}_{r,kn}^{online}(t_i) \tag{3.68}$$

The unknown part can then be computed as :

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}\,\mathbf{\Phi}_r\,\hat{\mathbf{a}}_r(t_i) = \mathcal{S}_{unkn}\,\mathbf{\Phi}_r\ (\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^{\dagger}\ \mathbf{u}_{r,kn}^{online}(t_i) \tag{3.69}$$

Let us define

$$\mathbf{A} = \mathcal{S}_{unkn}\,\mathbf{\Phi}_r\ (\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^{\dagger}\ \in \boldsymbol{R}^{M_{unkn}\times M_{kn}} \tag{3.70}$$

Then equation (3.69) becomes :

$$\mathbf{u}_{r,unkn}(t_i) = \mathbf{A}\,\mathbf{u}_{r,kn}^{online}(t_i) \tag{3.71}$$

The matrix $\mathbf{A}$ is computed only once for given operating conditions. Equation (3.71) is the simplest mathematical relation that can link a unknown information to the known information.

- **Remark 1**

  The accuracy of the IROM depends on the ratio between the number of known and unknown variables.

- **Remark 2**

  The accuracy of the IROM depends on the number of retained modes with respect to the number of known variables. An important requirement for the success of this method is that the number of retained modes should be less than the number of known information (sensors). This requirement is important to achieve a unique minimum for the problem (3.67).

- **Remark 3**

  The known information can include a physical quantity different from the field of interest. For instance, when computing the temperature field, we can add the cooling fluid temperature or the convection coefficient if they are available. The added information should preferably have a strong link with the field that we want to compute.

3.4 Conclusion

In this chapter, a survey on reduction methods have been conducted and the difference between reduction methods have been outlined. The chosen reduction method is based on the Singular Value Decomposition (SVD) for its convenience with commercial software.

The two reduction methods used in this work have been described and both can be seen as a combination of two stages: the learning stage and the usage stage.

In the learning stage a reduced basis is computed from FEM solution using SVD. This stage is the same for the two methods.

For the first method (direct ROM), the usage stage is based on Galerkin projection. It consists in projecting the FEM on the reduced basis which results in a low dimensional model. Simulations using the reduced model are much faster than with the FEM.

For the second reduction method (IROM), the usage stage is based on the fact that the known information (temperature from sensors) and the unknown information have the same coordinates in the reduced basis. Therefore, these coordinates will be computed from the known information (temperature provided by sensors), and then the unknown part can be determined.

In Chapters 4 and 5, an application of the two methods will be presented in the case of the industrial I2MPECT power electronic module.

Bibliography

[1] Vladimir Buljak. *Inverse analyses with model reduction: proper orthogonal decomposition in structural mechanics*. Springer Science & Business Media, 2011.

[2] Vladimir Buljak and Giulio Maier. Proper orthogonal decomposition and radial basis functions in material characterization based on instrumented indentation. *Engineering Structures*, 33(2):492–501, 2011.

[3] Vladimir Buljak. Proper orthogonal decomposition and radial basis functions algorithm for diagnostic procedure based on inverse analysis. *FME Transactions*, 38(3):129–136, 2010.

[4] Gabriella Bolzon and Vladimir Buljak. An effective computational tool for parametric studies and identification problems in materials mechanics. *Computational mechanics*, 48(6):675–687, 2011.

[5] D Xiao, F Fang, C Pain, and G Hu. Non-intrusive reduced-order modelling of the navier–stokes equations based on rbf interpolation. *International Journal for Numerical Methods in Fluids*, 79(11):580–595, 2015.

[6] Z Lin, D Xiao, F Fang, CC Pain, and Ionel M Navon. Non-intrusive reduced order modelling with least squares fitting on a sparse grid. *International Journal for Numerical Methods in Fluids*, 83(3):291–306, 2017.

[7] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[8] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[9] GML Gladwell. Branch mode analysis of vibrating systems. *Journal of sound and vibration*, 1(1):41–59, 1964.

[10] Alain Neveu, Khalil El-Khoury, and Bernard Flament. Simulation de la conduction non linéaire en régime variable: décomposition sur les modes de branche. *International journal of thermal sciences*, 38(4):289–304, 1999.

[11] Olivier Quéméner, Frédéric Joly, and Alain Neveu. The generalized amalgam method for modal reduction. *International Journal of Heat and Mass Transfer*, 55(4):1197–1207, 2012.

[12] Olivier Quemener, Alain Neveu, and Etienne Videcoq. A specific reduction method for the branch modal formulation: Application to a highly non-linear configuration. *International journal of thermal sciences*, 46(9):890–907, 2007.

[13] Yu-Ching Yang and Wen-Lih Chen. A nonlinear inverse problem in estimating the heat flux of the disc in a disc brake system. *Applied Thermal Engineering*, 31(14-15):2439–2448, 2011.

[14] SR Carvalho, SMM Lima e Silva, AR Machado, and G Guimaraes. Temperature determination at the chip–tool interface using an inverse thermal model considering the tool and tool holder. *Journal of Materials Processing Technology*, 179(1-3):97–104, 2006.

[15] Olivier Quéméner, Frédéric Joly, and Alain Neveu. On-line heat flux identification from a rotating disk at variable speed. *International Journal of Heat and Mass Transfer*, 53(7-8):1529–1541, 2010.

[16] Etienne Videcoq, Myriam Lazard, Olivier Quemener, and Alain Neveu. Online temperature prediction using a branch eigenmode reduced model applied to cutting process. *Numerical Heat Transfer, Part A: Applications*, 55(7):683–705, 2009.

[17] D Petit and R Pasquetti. Réduction de modèle par identification de modes dominants: application à un modèle bidimensionnel de diffusion thermique. *Revue de physique appliquée*, 25(8):831–842, 1990.

[18] D Petit, R Hachette, and D Veyret. A modal identification method to reduce a high-order model: Application to heat conduction modelling. *International Journal of Modelling and Simulation*, 17(3):242–250, 1997.

[19] D Petit and R Hachette. Model reduction in linear heat conduction: use of interface fluxes for the numerical coupling. *International journal of heat and mass transfer*, 41(21):3177–3189, 1998.

[20] Yassine Rouizi. *Réduction de modèle par identification en convection forcée pour des systèmes soumis à des conditions aux limites thermiques instationnaires: application à l'écoulement le long d'une marche avec contrôle thermique par retour d'état*. PhD thesis, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique-Poitiers, 2010.

[21] Manuel Girault and Daniel Petit. Identification methods in nonlinear heat conduction. part i: Model reduction. *International Journal of Heat and Mass Transfer*, 48(1):105–118, 2005.

[22] Manuel Girault and Daniel Petit. Identification methods in nonlinear heat conduction. part ii: inverse problem using a reduced model. *International Journal of Heat and Mass Transfer*, 48(1):119–133, 2005.

[23] Y Rouizi, M Girault, Y Favennec, and D Petit. Model reduction by the modal identification method in forced convection: Application to a heated flow over a backward-facing step. *International Journal of Thermal Sciences*, 49(8):1354–1368, 2010.

[24] Etienne Videcoq, Alain Neveu, Olivier Quemener, Manuel Girault, and Daniel Petit. Comparison of two nonlinear model reduction techniques: the modal identification method and the branch eigenmodes reduction method. *Numerical Heat Transfer, Part B: Fundamentals*, 49(6):537–558, 2006.

[25] Olivier Balima, Yann Favennec, Manuel Girault, and Daniel Petit. Comparison between the modal identification method and the pod-galerkin method for model reduction in nonlinear diffusive systems. *International Journal for numerical methods in engineering*, 67(7):895–915, 2006.

[26] Eric James Grimme. *Krylov projection methods for model reduction*. PhD thesis, Citeseer, 1997.

[27] Athanasios C Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Siam, 2005.

[28] Gilles Serre. *Construction de modèles réduits numériques pour les écoulements compressibles linéarisés*. PhD thesis, Ecole Centrale de Lyon, 2012.

[29] Pierre Ladevèze, J-C Passieux, and David Néron. The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21-22):1287–1296, 2010.

[30] Francisco Chinesta, Amine Ammar, Adrien Leygue, and Roland Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011.

[31] Amine Ammar, Bechir Mokdad, Francisco Chinesta, and Roland Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of non-Newtonian fluid Mechanics*, 139(3):153–176, 2006.

[32] Francisco Chinesta, Amine Ammar, and Elías Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational methods in Engineering*, 17(4):327–350, 2010.

[33] Pierre Ladeveze and Anthony Nouy. On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30):3061–3087, 2003.

[34] Francisco Chinesta, Amine Ammar, and Elías Cueto. Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, 83(8-9):1114–1132, 2010.

[35] David González, Amine Ammar, Francisco Chinesta, and Elías Cueto. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81(5):637–659, 2010.

[36] Ch Ghnatios, Françoise Masson, Antonio Huerta, Adrien Leygue, Elías Cueto, and Francisco Chinesta. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213:29–41, 2012.

[37] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations: a primer*. Springer Science & Business Media, 2013.

[38] Enrique Nadal, Francisco Chinesta, Pedro Díez, FJ Fuenmayor, and FD Denia. Real time parameter identification and solution reconstruction from experimental data using the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 296:113–128, 2015.

[39] Antoine Dumon, Cyrille Allery, and Amine Ammar. Simulation of heat and mass transport in a square lid-driven cavity with proper generalized decomposition (pgd). *Numerical Heat Transfer, Part B: Fundamentals*, 63(1):18–43, 2013.

[40] Xi Zou, Michele Conti, Pedro Díez, and Ferdinando Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2):230–251, 2018.

[41] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[42] Hugh Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3):399–417, 1963.

[43] Yuri Evtushenko. Generalized lagrange multiplier technique for nonlinear programming. *Journal of Optimization Theory and Applications*, 21(2):121–135, 1977.

[44] Chris Homescu, Linda R Petzold, and Radu Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis*, 43(4):1693–1714, 2005.

[45] Markus Müller. *On the POD method: an abstract investigation with applications to reduced-order modeling and suboptimal control*. PhD thesis, 2008.

[46] David Amsallem, Julien Cortial, Kevin Carlberg, and Charbel Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *International journal for numerical methods in engineering*, 80(9):1241–1258, 2009.

[47] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal*, 46(7):1803–1813, 2008.

[48] Thuan Lieu and Michel Lesoinne. Parameter adaptation of reduced order models for three-dimensional flutter analysis. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*, page 888, 2004.

[49] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[50] Karen Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.

[51] Richard Everson and Lawrence Sirovich. Karhunen–loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[52] Tan Bui-Thanh, Murali Damodaran, and Karen E Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal*, 42(8):1505–1516, 2004.

[53] T Bui-Thanh, Murali Damodaran, and Karen Willcox. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. In *21st AIAA Applied Aerodynamics Conference*, page 4213, 2003.

[54] Nathan E Murray and Lawrence S Ukeiley. An application of gappy pod. *Experiments in Fluids*, 42(1):79–91, 2007.

Chapter 4

POD/Galerkin reduced order model of one leg power electronic module

Sommaire

4.1 Introduction

Failure occurring in electrical power package are mainly due to excessive temperature amplitude or cycling [1]. Hence, the reliability of electronic components is strongly dependent on the temperature. To avoid failure in the electronic package, we should choose the suitable cooling solution, design it accurately and monitor on real time the temperature of electronic components.

An accurate thermal design of the cooling system requires an accurate thermal model of the system (power electronic module and the cooling system). Many thermal models have been worked out in the literature and are based on a lumped parameter representation of the system [2–7]. This lumped parameter representation is a thermal network consisting of thermal resistances and capacitances. The resistances and capacitances can either be computed from physical and geometrical properties of the system or can be identified from results.

For instance, M. Ishizuka et al [8] constructed a transient thermal network of an electronic device by assuming one-dimensional heat conduction. To take into account the three-dimensional effect of heat conduction, F. N. Masana [9] derived an expression of the spreading resistance and validated it by comparison with an analytical solution. The weakness of the thermal spreading resistance lies in the fact that it varies with operating conditions (heat transfer coefficient and power losses density) and the power module layout [10]. Therefore, this model is not suitable for the heat transfer in multilayer electronic packages.

Another method for deriving compact thermal models is based on the identification of the thermal network parameters from reference results (temperature obtained by finite element simulation or by experiments). A straightforward explanation of this approach can be found in [11]. This method generates a thermal network that depends solely on the power losses and a reference temperature [12]. This thermal network model does not depend explicitely on the cooling system design parameters (heat transfer coefficient). Therefore, it is not convenient for the design of the cooling system.

The most accurate thermal models are those obtained by space discretization methods (finite element or finite difference methods, ...). The main drawback of the latter methods is its high computing time when dealing with high-dimensional systems requiring a large number of mesh nodes. This weakness makes the FEM unsuitable for realtime monitoring of the temperature or for the design of the electronic package.

In this chapter, we propose a model that gathers the mathematical structure of a compact thermal model and the accuracy of a finite element model. These two qualities (compact mathematical structure and accuracy) are achieved by proper orthogonal decomposition and Galerkin projection.

In the first section the studied system is described in details and the considered boundary conditions are adressed. Then, the convergence properties of the FEM are depicted. After that, the steps of the ROM construction are addressed. In the following sections, the ROM has been studied for the steady and transient states. The learning stage of each model is described. The performances of the steady and transient ROMs have been assessed when the usage operating conditions are different from those of the learning conditions. The effect of the snapshots time step on the accuracy of the transient ROM is analyzed.

4.2 The studied power electronic module (PEM)

The studied industrial system consists of a three-legs power electronic module mounted on a cooling system. For the sake of simplicity, a single leg power module was considered in this study. The constructed model can be expanded to study the thermal behavior of the three legs power module by considering relevant values of the boundary conditions (the fluid temperature and the convection coefficient). Under the assumption that all heat flows to the heat sink air stream and not to the surroundings, the fluid temperature of the considered one leg power module T_f^{i+1} can be computed from the fluid temperature of the upstream one leg power module T_f^i by the following relationship:

$$T_f^{i+1} = T_f^i + \frac{P_i}{\dot{m}\, C_p}$$

where P_i, $\dot{m}$ and C_p are respectively the power losses in the i-th power module ($i = 1, 2, 3$ for a three phase DC/AC converter), the mass flow rate and the heat capacity of the cooling fluid.

4.2.1 Geometry and materials of the PEM

The power module described in the Chapter 1 consists of 12 MOSFETs. These 12 MOSFETs are soldered on a DBC. The DBC is mounted on a base plate in order to spread out the hot spots created by the power losses in MOSFETs. In Figure 4.1, the top face of the power module is presented with the corresponding dimensions in mm. The dimensions of the MOSFET chip are 6.44 mm $\times$ 4 mm $\times$ 0.19 mm.

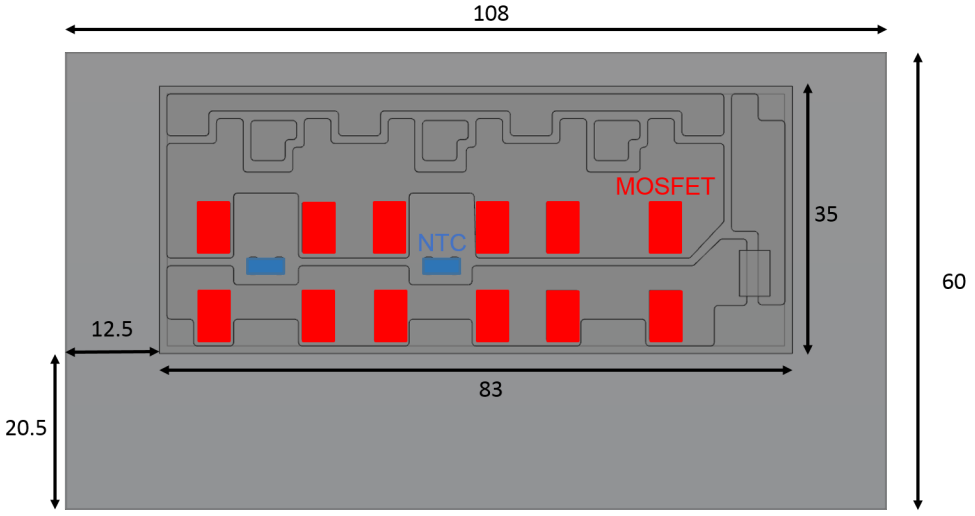


Figure 4.1 – The power electronic module top surface

A description of the power module layers is given in Figure 4.2.

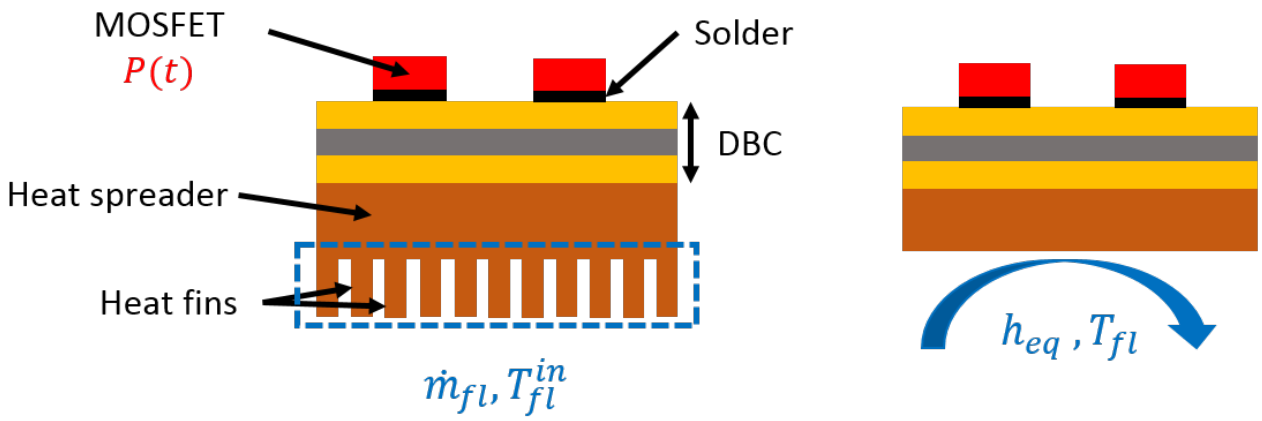


Figure 4.2 – The power electronic module layers and the equivalent heat transfer coefficient

The DBC consists of two copper layers and a ceramic layer. Each of the copper layers is 0.3 mm thickness. The ceramic layer is 0.35 mm thickness. The heat spreader is a copper base-plate measuring 3 mm thickness.

The physical properties of the used material in the power module are listed in Table 4.1

	ρ (kg/m^3)	C_p (J/kg/K)	k(W/m/K)
Copper	8700	385	400
AlN (Ceramic)	3255	800	180
SiC	3200	750	370

Table 4.1 – Physical properties of the materials used in the electronic power module

Each MOSFET is soldered to the DBC. The contact between the MOSFET and the DBC is not perfect which results in a contact thermal resistance. The contact thermal resistance between the MOSFET and the DBC is chosen to be equal to :

$$R_{solder}^{contact} = 3.10^{-5}\ K.m^2/W$$

The contact thermal resistance between the DBC and the heat spreader is chosen to be equal to :

$$R_{DBC/HS}^{contact} = 4.10^{-4}\ K.m^2/W$$

These values of the contact thermal resistances are chosen with respect to experimental measurements carried out on a single chip power module delivered by I2MPECT partners. The chosen contact thermal resistances are in the range of values found in the literature [13].

4.2.2 Boundary conditions and thermal load

The top surface of the power module is covered by an electrical insulator which is also a thermal insulator. Therefore an adiabatic boundary condition has been considered on this surface. The edge surface of the power module are surrounded by a plastic border. For that reason, an adiabatic boundary condition has been considered on this surface. The bottom surface of the power module is in contact with the cooling system as shown in Figure 4.2. Consequently a

convection boundary condition is considered here. The choice of the heat coefficient and the temperature is goverened by heat transfer law in the heat sink as described in Appendix A. In the case of steady state ROM many values of the heat coefficient have been considered and the accuracy of the ROM with respect to this parameter have been studied. In the case of transient ROM, the heat coefficient and the fluid temperature are chosen equal to $h_{eq} = 4000\ W/m^2/K$ and $T_f = 50\ °C$. The inlet fluid temperature $T_f = 50\ °C$ corresponds to the operating conditions that are defined by industrial partners.

Power losses occur in MOSFET chips. We assume that this dissipated power is uniformily distributed over the MOSFET volume. In the case of steady state ROM, several values of power losses have been considered in order to assess the performance of the ROM. In the case of transient ROM, we considered a power loss profile depending on time and the accuracy of the ROM with respect to this parameter is investigated.

4.3 Convergence of the finite element model (FEM)

4.3.1 Mesh convergence

In order to assess the discretization quality using finite element method, a convergence study has been carried out. In this study, 5 meshes with different refinement level have been considered. For each mesh, the maximum temperature over the power module is computed and the energy conservation is checked. The energy conservation condition that has been checked can be written as :

$$P - h_{eq}\, S\, (T_s - T_f) = 0$$

where S in the bottom surface of the power module submitted to convection heat transfer, T_s is its mean temperature, and P is the overall power losses in MOSFETs. The mesh convergence study have been performed in the case of steady state heat transfer in order to avoid time discretization errors. Table 4.2 summarizes the mesh convergence results in the case of linear shape functions. $\mathbf{h_{max}}$ and $\mathbf{h_{min}}$ stands for the maximum and minimum mesh size. The convergence study have been performed in the case of $h_{eq} = 4000\ W/m^2/K$, $T_f = 50\ °C$ and $P = 300\ W$.

	Number of nodes	$\mathbf{h_{min}(mm)}$	$\mathbf{h_{max}(mm)}$	$T_{max}(°C)$	Energy balance
Mesh 1	505818	0.009	2.8	153	0
Mesh 2	142240	0.064	4.6	152.91	0
Mesh 3	19984	0.056	5.7	152.48	0
Mesh 4	8386	0.177	9.2	151,8	0
Mesh 5	2953	0.649	15.2	149.26	0

Table 4.2 – Mesh convergence in the case of linear shape function

It can be noticed that the energy conservation is verified for all meshes. The maximum temperature (T_{max}) obtained for the finest mesh is equal to $153\ °C$.

A good trade-off between small number of nodes and an accurate model is made by choosing the mesh 3 with linear shape functions. This choice results in an absolute error on the maximum temperature equal to $0.5\ °C$ at steady state.

4.3.2 Temporal convergence

The effect of the time step has been studied. The transient solution has been computed using equation (2.28) which is an analytical solution for the time integration problem. The drawback of the analytical solution is the high computing time. Therefore it is recommended to use a numerical integration scheme. The chosen time integration scheme is the Crank-Nicolson scheme. Four time steps have been considered: $\Delta t = 0.001\ s$, $\Delta t = 0.01\ s$, $\Delta t = 0.1\ s$ and $\Delta t = 1\ s$. The absolute error time evolution in the node with the biggest error for these time steps has been plotted on Figure 4.3.

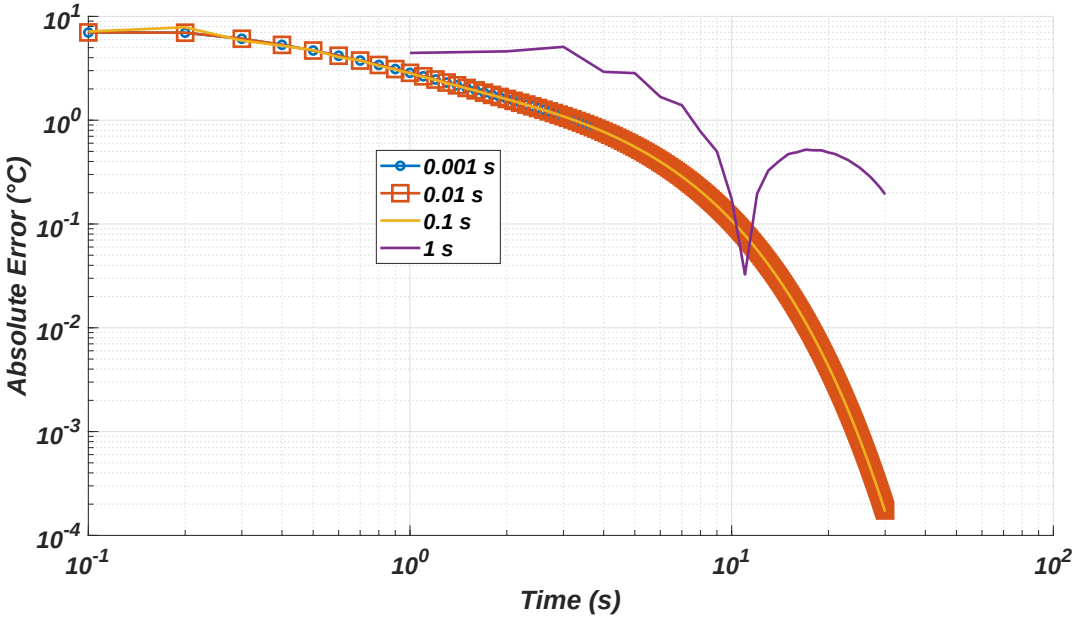


Figure 4.3 – Time step convergence in the node of the highest error

The time step convergence study has been carried out in the case of $h_{eq} = 4000\ W/m^2/K$, $T_f = 60\ °C$ and $P = 400\ W$.

The obtained results show that a time step convergence is achieved for $\Delta t = 0.1\ s$. This time step will be used in the next sections.

4.4 Construction of the reduced order model

The main steps to construct the reduced order model are:

1. **Construction of the finite element model (FEM) on $COMSOL^{®}$:**
   - Constructing of the power module geometry.
   - Assigning the material for each component of the geometry.
   - Setting the boundary conditions.
   - Constructing the mesh: in this task many meshes are assessed and the best one is kept.

- Running time-dependent simulation for given operating conditions.

2. **Recovering the FEM matrices:**

   - The mass matrix **C** does not depend on boundary conditions, thus it is recovered from the previous FEM.
   - The stiffness matrix depends on boundary conditions ($\mathbf{K}(h) = \mathbf{K}_0 + h\,\mathbf{K}_{cv}$ as described in equation (2.22)). Therefore, two finite element simulations are run. In the first one, by setting $h = 0$ in $COMSOL^{®}$, the matrix $\mathbf{K}_0$ is recovered. In the second one, we choose for instance $h = 1W/m^2/K$ and $\mathbf{K}(h = 1)$ is recovered, hence $\mathbf{K}_{cv} = \mathbf{K}(h = 1) - \mathbf{K}_0$.
   - The load vector depends on boundary conditions ($\mathbf{l}(P(t), h, T_{ext}) = P(t).\mathbf{l}_Q + h\,T_{ext}\,\mathbf{l}_{cv}$ as described by equation (2.23)). Another two finite element simulations are carried out in order to compute $\mathbf{l}_Q$ and $\mathbf{l}_{cv}$. In the first simulation, we set $P = 0$, $h = 1$ and $T_{ext} = 1$. In this case the vector $\mathbf{l}_{cv}$ is recovered. In the second simulation we set $P = 1$, $h = 0$ and $T_{ext} = 0$ and the vector $\mathbf{l}_Q$ is recovered.
   - After recovering finite element matrices, a time-dependent simulation for arbitrary boundary conditions is carried out using $MATLAB^{®}$ and $COMSOL^{®}$. Obtained result showed good agreement between the two computational tools which validates the procedure used to recover finite element matrices.
   - Assess the temporal convergence of the finite element model by testing various values of time steps.

3. **Preparing matrices used by the ROM:**

   - Run finite element simulation(s) on $MATLAB^{®}$ for given operating conditions $(P(t), h, T_{ext})$ in order to construct the snapshots matrix.
   - Apply SVD on the snapshots matrix and save the dominant POD modes ($\mathbf{\Phi}_r$).
   - Construct the matrices given by equations (3.55).

4. **Online computing of the ROM**

   - Compute the ROM matrices (equation (3.56)) for given operating conditions.
   - Compute the initial condition of the ROM as given in equation (3.57).
   - Integrate in time the ROM given by equation (3.58a).
   - Compute the temperature field from equation (3.58c).

4.5 Steady state reduced order model

When designing a cooling system for a given power electronic module, the main design criterion is based on the maximum temperature. When the power losses are constant, the maximum temperature is reached at the steady state. A cooling solution is specified by the value of the heat coefficient. A handy procedure to assess accurately and quickly the effectiveness of a cooling system is to use a steady state thermal reduced order model. In this section, the construction steps of the steady state reduced order model are depicted and the performance of the ROM (accuracy and computing time) are investigated.

4.5.1 Learning stage: POD

The first step is to compute the temperature field for a set of operating conditions (h, P, T_f). The used finite element mesh consists of $M = 19984$ nodes. For instance, the temperature field is presented on Figure 4.4 in the case of $P = 200\ W$, $h_{eq} = 1000\ W/m^2/K$ and $T_f = 50\ °C$.

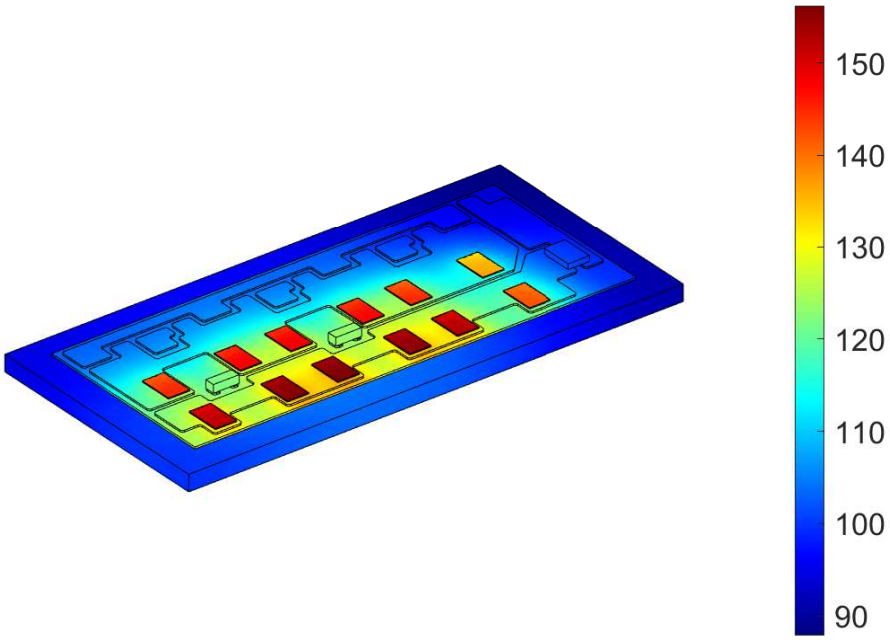


Figure 4.4 – Temperature field in the case of $P = 200\ W$, $h_{eq} = 1000\ W/m^2/K$ and $T_f = 50\ °C$

The temperature fields will be used to construct the snapshots matrix. The chosen parameters used to construct the snapshots matrix are presented on Figure 4.5. The fluid temperature is equal to $50\ °C$ and is kept the same for all cases.

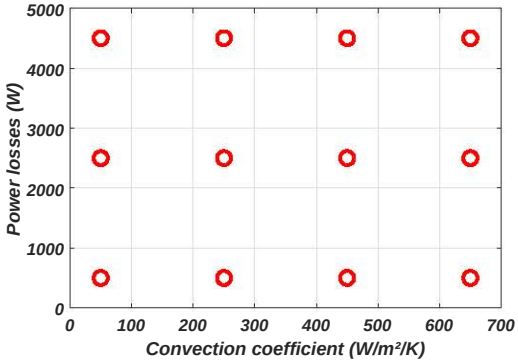
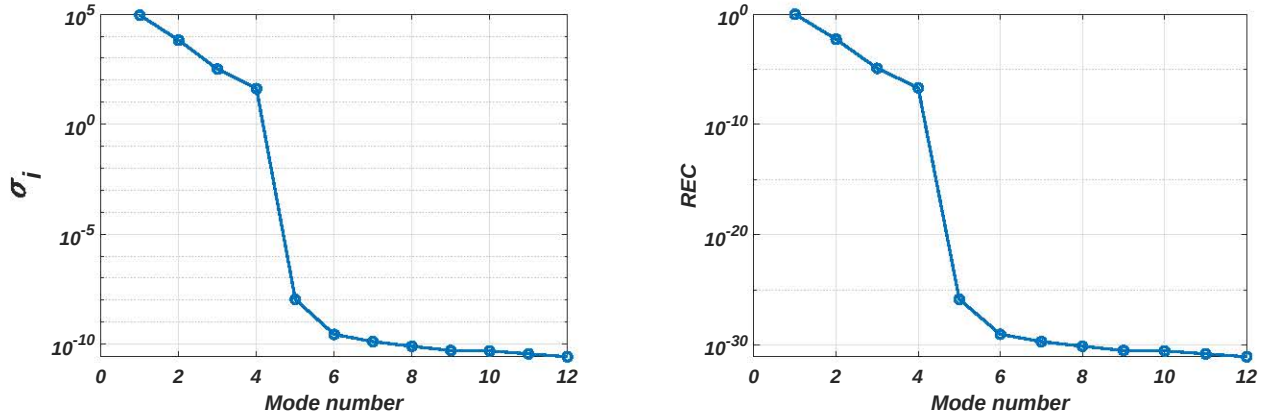


Figure 4.5 – The Learning map

The snapshots matrix is constructed from the computed temperature fields. Each column of the snapshot matrix correspond to the temperature field obtained for the pair (h_{eq}, P). The obtained snapshots matrix is $M \times N$ where M is the number of nodes and $N = 12$ is the number of pair (h_{eq}, P). The singular value decomposition is performed on the snapshots matrix. The first 6 spatial modes of this decomposition are presented on Figure 4.7. The singular value distribution and the relative energy content are presented on Figure 4.6.

Figure 4.7 shows that the higher order modes have higher spatial gradients.

It can be seen from Figure 4.6 that the singular values decreases drastically. This rapid decrease is a typical behavior of the singular value decomposition. If we want to keep the modes having a relative energy content greater than 10^{-10}, then we should retain 4 modes. We can consider the singular value distribution to select the dominant modes. It has been shown in Chapter 3 that the reconstruction error using the first r modes is bounded by $\sqrt{min(M,N) - r}\,\sigma_{r+1}$ (see equation (3.37)). Therefore, the reconstruction error (based on maximum norm) induced by taking $r = 4$ is bounded by $\sqrt{12-4}\;10^{-8} = 2.8\;10^{-8}\;°C$. During the usage stage that will be addressed in the following section, the coordinates of a new solution will be computed in this POD basis. This approach reduces the number of unknowns from $M = 19984$ to $r = 4$.



Singular values Relative Energy Content (REC)

Figure 4.6 – Singular values and REC distribution

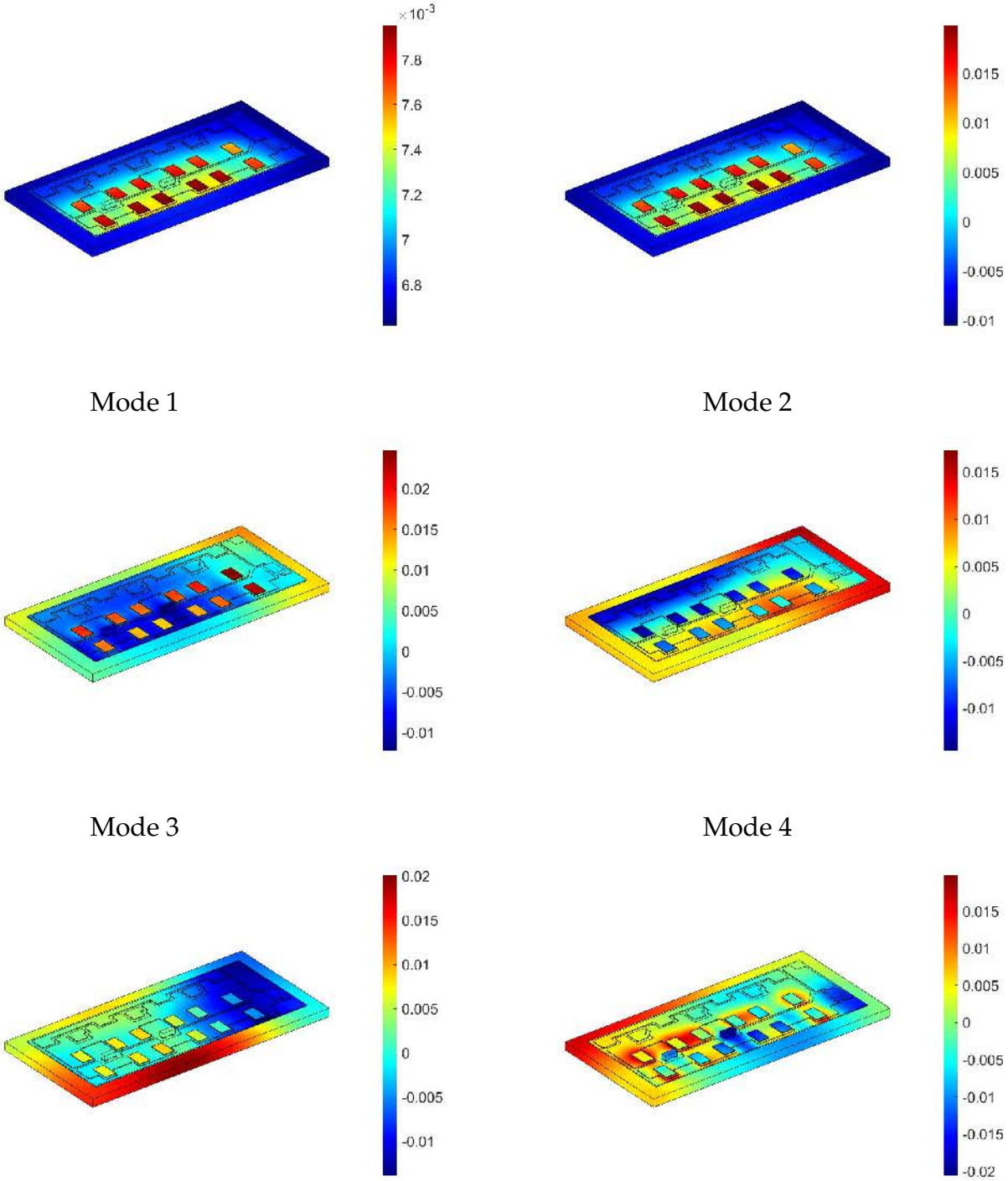


Figure 4.7 – The 6 first POD modes

4.5.2 Usage stage

During this stage, the solution will be computed in the reduced basis (4 first POD modes). In order to assess the robustness of this approach, we consider operating conditions (h, P, T_f) that can be identical (red circles on Figure 4.8)or different from the learning stage (Figure 4.5). The usage stage operating conditions are presented on Figure 4.8. The temperature field has been computed for all the operating conditions (117 pair (h, P, T_{fl})) of Figure 4.8 using the FEM and the ROM.

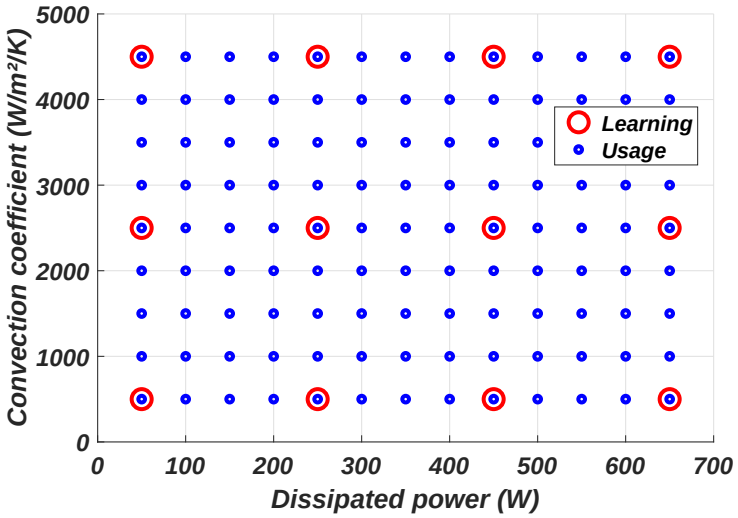
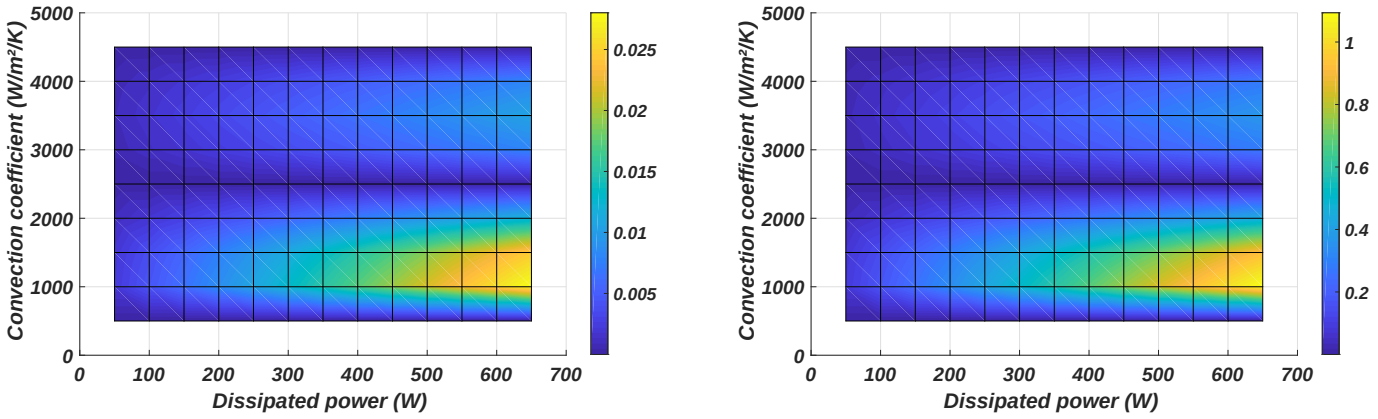


Figure 4.8 – The Learning and usage map

The difference between the FEM and the ROM is computed and its norms are calculated and plotted on Figure 4.9 for each usage operating condition.



maximum norm $l^2 - norm$

Figure 4.9 – Error distribution in the case of 4 retained modes

It can be noticed from Figure 4.9 that the error is important for operating condition different from the ones considered during the learning stage. This behavior can be explained by the fact

that snapshots skipped during learning (with respect to usage) are not well presented by the POD basis. It can be seen that this error increases when the dissipated power increases and the convection coefficient decreases.

In order to assess the selection criteria of dominant modes, we plot on Figure 4.10 the absolute error field for different number of retained modes. The chosen operating condition is : $P = 500\,W$ and $h_{eq} = 2000\,W/m^2/K$. It can be seen from Figure 4.10 that when increasing the number of retained modes, the absolute error field decreases. The steady state can not be described by a single mode. Using only the first mode results in an important error that is localised on the heat sources (MOSFETs), and adding the second mode allows to remove it. It can be seen that far $r > 4$, no improvement in the accuracy is notable which proves the robustness of the dominant modes selection criteria.

Now, this study will be performed for all usage operating conditions. The l^2 norm of the difference between the FEM and the ROM will be plotted for many values of retained modes. Obtained results are presented on Figure 4.11. We can see that for the ROM of a single POD mode, the error increases when power losses increases. This behavior does not depend on the difference between learning and usage operating conditions.

Adding modes 2, 3 and 4, decreases the magnitude of the previous error without any dependence on the dissipated power or the convection coefficient. From $r = 4$, the error in not affected by adding other modes and the observed behavior is due to the difference between the learning and the usage operating conditions.

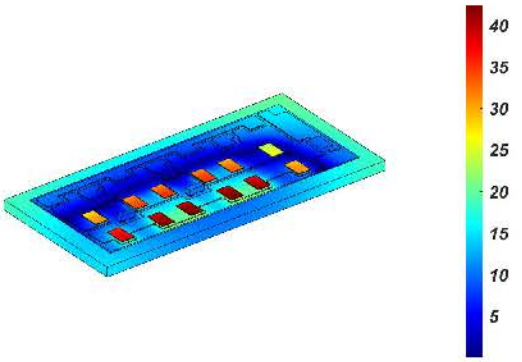
We have seen that a ROM of 4 modes results in an l^2 norm error that does not exceed $1\ ^\circ C$ and an absolute error less than $10^{-2}\ ^\circ C$.

The computing time for a single simulation for both finite element model and ROM is given in the Table 4.3.

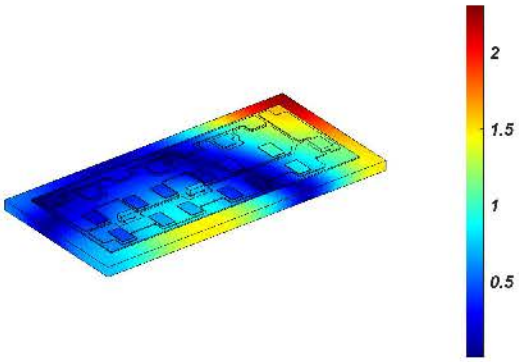
Model	CPU time (s)
FEM	0.267
ROM	0.001

Table 4.3 – Computing time

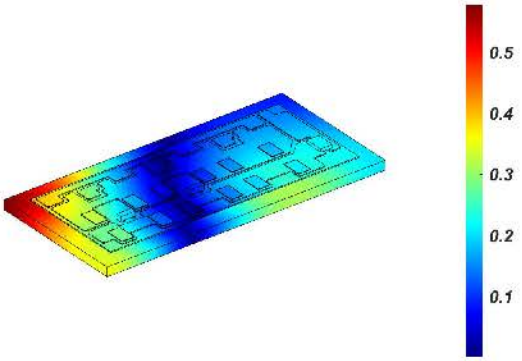
Thanks to the ROM a gain of 267 is achieved. Such computing time, allows to reduce drastically the design phase of a cooling system for large range operating conditions.



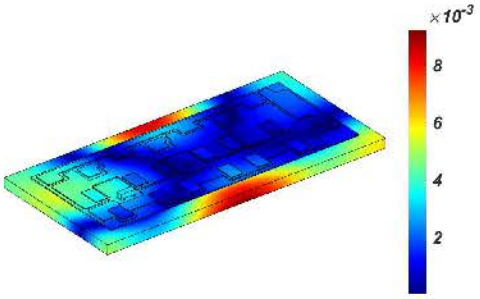
r=1



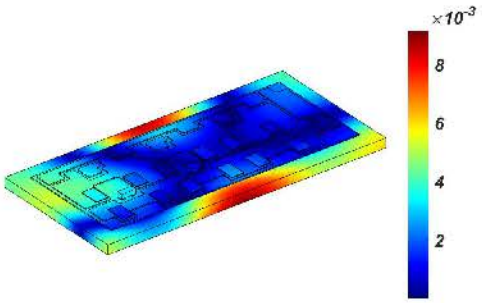
r=2



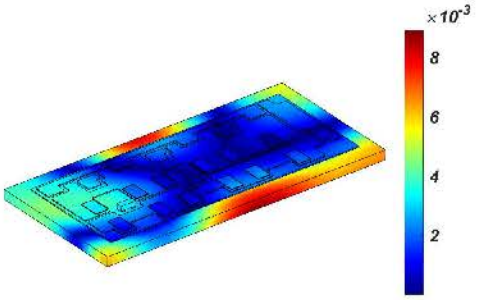
r=3



r=4



r=5



r=6

Figure 4.10 – Absolute error $|\mathbf{T}_r(x) - \mathbf{T}_{MEF}(x)|$ in the case of $P = 500\,W$ and $h_{eq} = 2000\,W/m^2/K$ for different number of retained modes

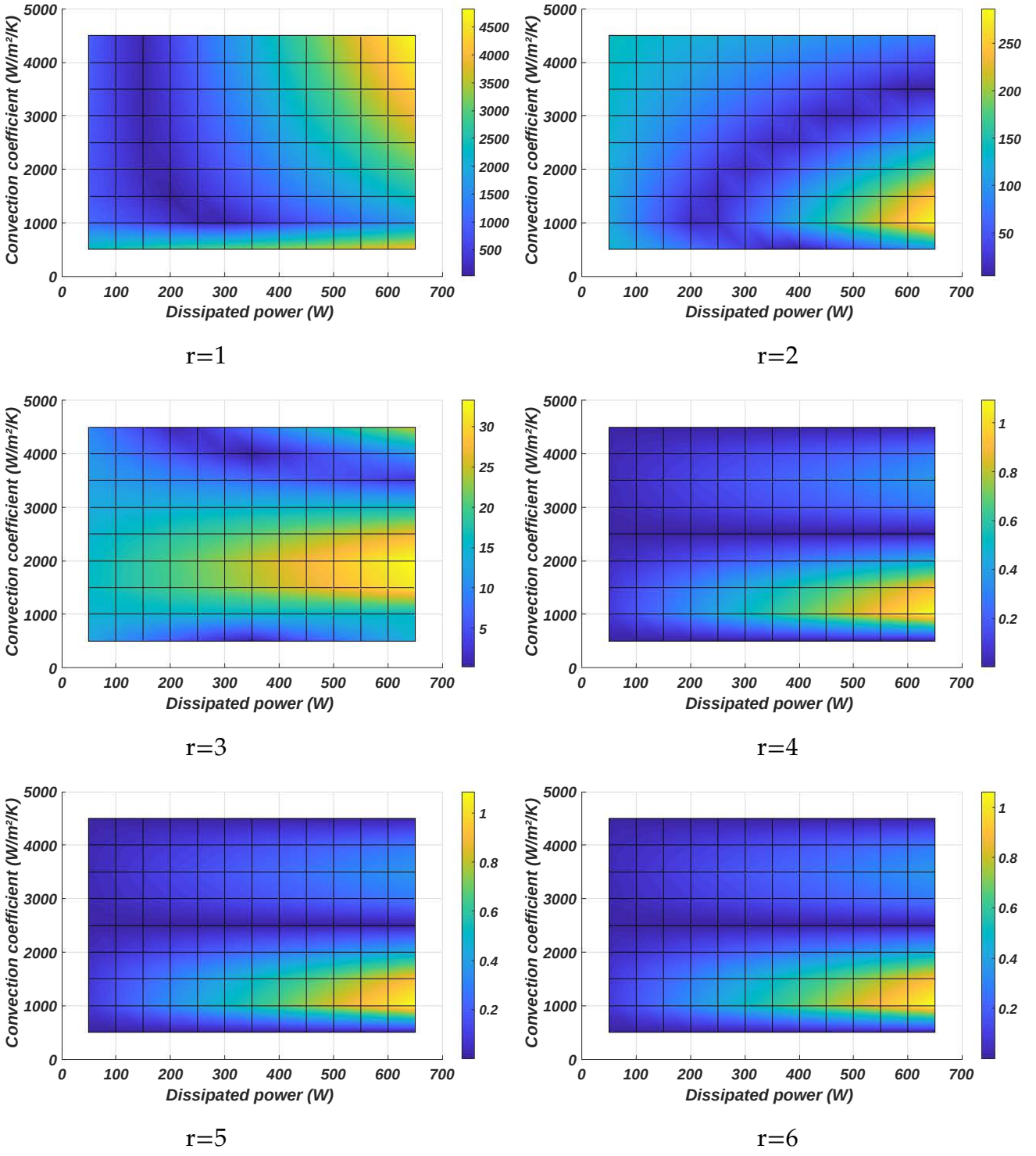


Figure 4.11 – The error (l^2 norm) map for different number of retained modes

4.6 Transient reduced order model

As described in Chapter 1, the reliability of electronic components depends on temperature. In order to increase the electronic components lifetime and prevent failure, the temperature in the power module should be tracked in real time. Real time monitoring can be achieved by putting a temperature sensor on the electronic component. Unfortunately, such procedure is not allowed in I2MPECT project for packaging reasons. The proposed solution is to construct a virtual sensor that will perform the same task as a real temperature sensor. This virtual sensor is the thermal reduced order model. The ROM should be accurate and fast in order to measure in real time the temperature. In this section, the construction steps of the transient ROM will be presented and its performance will be assessed.

4.6.1 Learning stage: spatial modes and eigen values distribution

The first step to construct the ROM, is to perform a transient simulation for given operating conditions. The boundary conditions considered in the learning stage are : $h = 4000\ W/m^2/K$ and $T_f = 50\ °C$. The initial condition considered for this simulation is a uniform temperature field equal to $T_0 = 20\ °C$. The power profile used for learning is presented on Figure 4.12. The sudden variation in time of the power profile is a typical behavior in real operational conditions. During the first $25\ s$, the power is set to $0\ W$ in order to adjust to the effect of boundary conditions. From $25\ s$ to $50\ s$ the power is set to $400\ W$ in order to involve the power losses effect in the snapshots matrix.

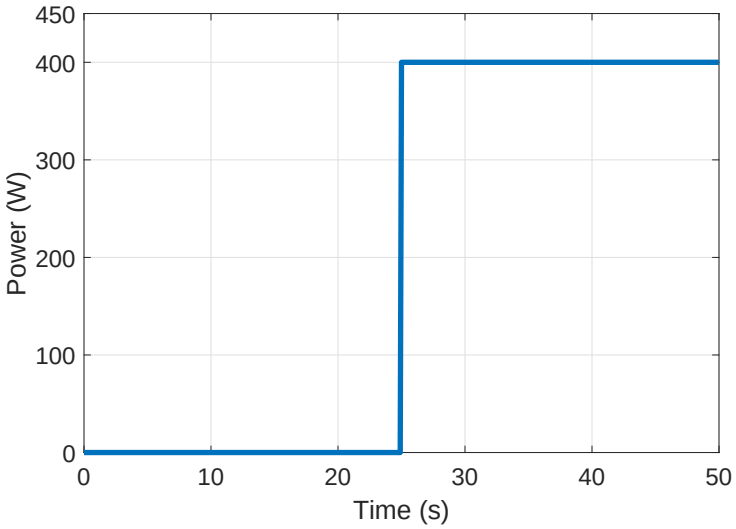
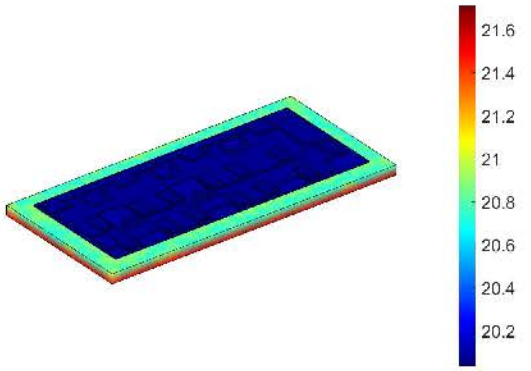
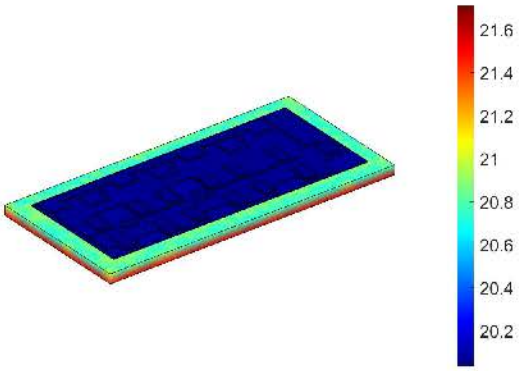


Figure 4.12 – The Learning power profile

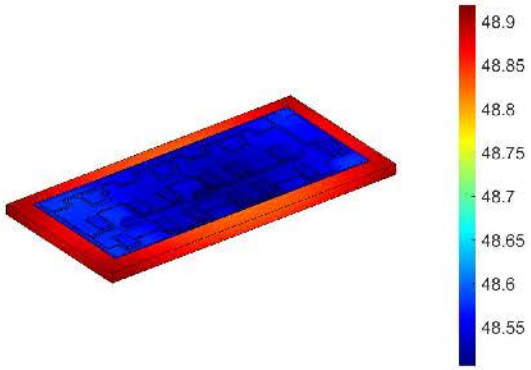
The obtained temperature field is stored in the snapshots matrix. The time step separating two consecutive snapshots is equal to $0.1\ s$ which is the FEM time step. The finite element mesh consists of $M = 19984$ nodes. Therefore, the snapshots matrix $\mathbf{U}$ belongs to $\mathbb{R}^{19984\times501}$. For instance, 6 snapshots at instants $t = 0.1\ s$, $t = 1\ s$, $t = 10\ s$, $t = 20\ s$, $t = 30\ s$ and $t = 40\ s$ are presented on Figure 4.13.



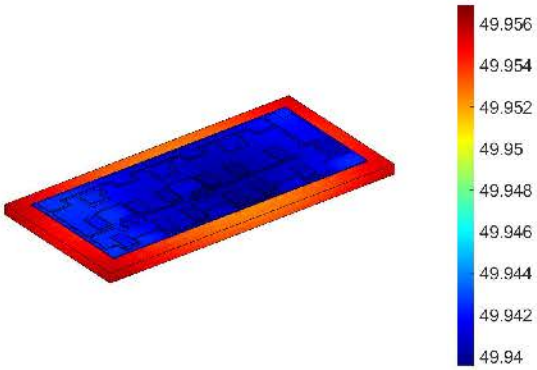
$t = 0.1\ s$



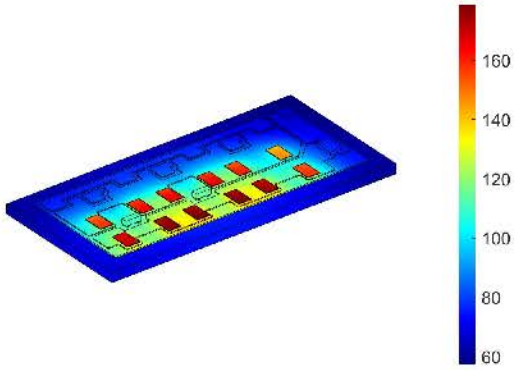
$t = 1\ s$



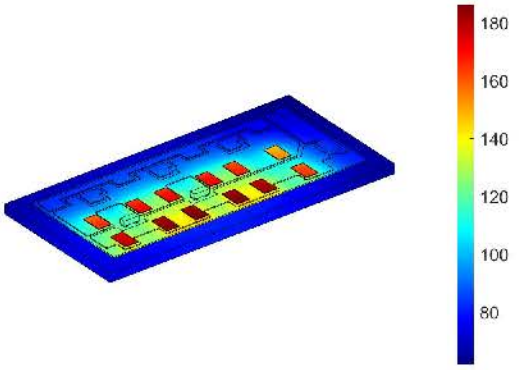
$t = 10\ s$



$t = 20\ s$



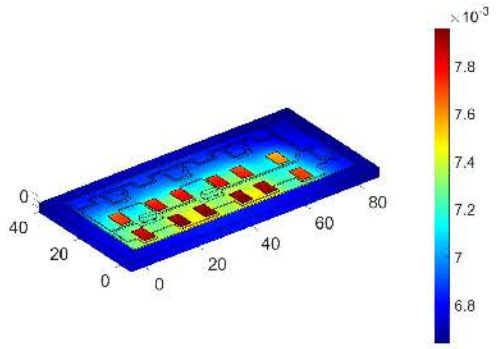
$t = 30\ s$



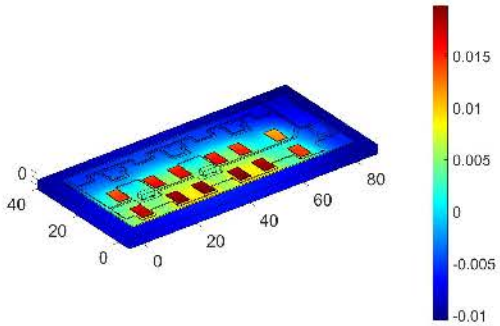
$t = 40\ s$

Figure 4.13 – The temperature field at different time instants

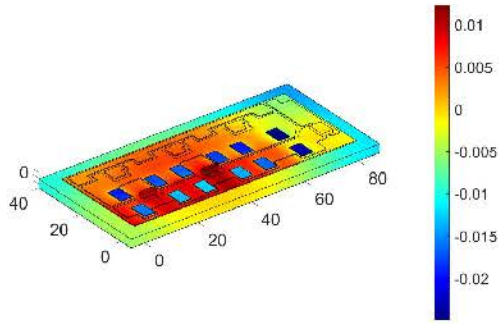
The SVD is applied on the snapshots matrix. The first 8 POD modes are presented on Figure 4.14.



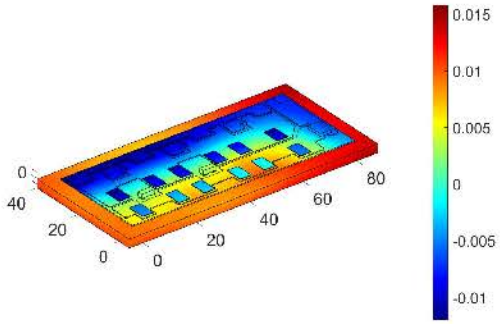
Mode 1



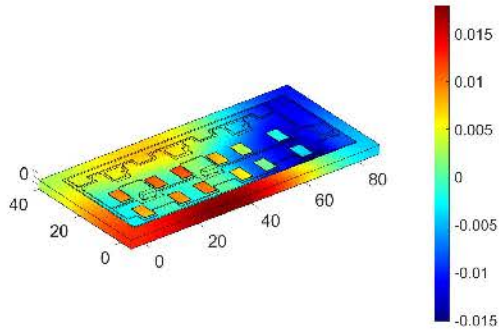
Mode 2



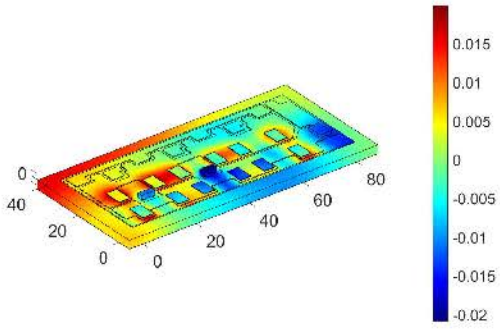
Mode 3



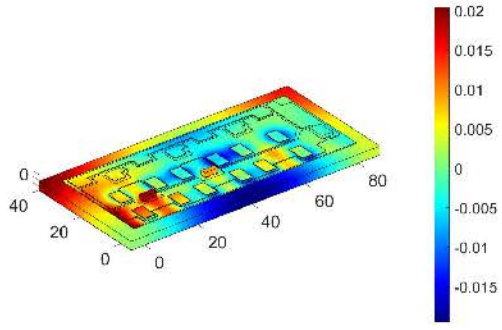
Mode 4



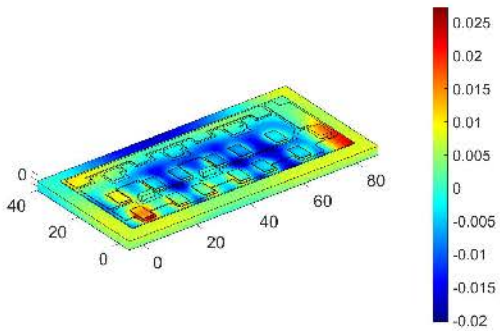
Mode 5



Mode 6



Mode 7



Mode 8

Figure 4.14 – The first 8 POD modes

Figure 4.14 shows that higher order modes have larger spatial gradients.
In Figure 4.15, temporal modes are presented.

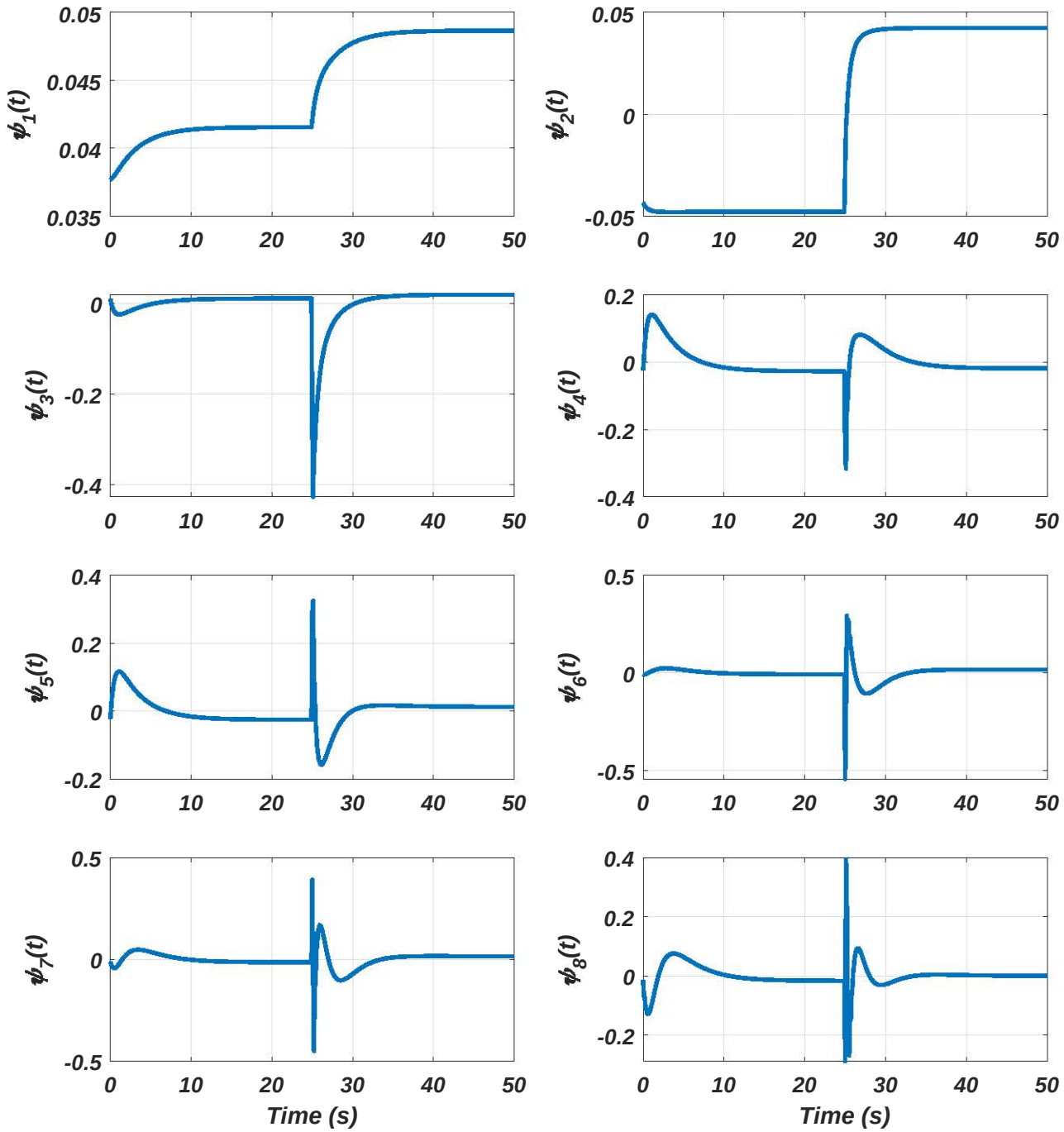


Figure 4.15 – The first 8 temporal modes

The temporal behavior of the temperature field is included in the temporal modes. When the power losses vary suddenly (around $25\ s$) in time, the temporal modes vary as well. We can notice that the time variation of the high order modes is much more important than the first ones. Such behavior implies that important time variation of the temperature field are captured by high order modes. In order to confirm this purpose, we plot on Figure 4.16 and 4.17, the 10 first temporel modes for $0\ s \leq t \leq 25\ s$ and for $25\ s < t \leq 50\ s$. A logarithmic time scale is used in order to better visualize important time variation. The temporal modes variation when $0\ s \leq t \leq 25\ s$ are due to the difference between the initial temperature and the fluid temperature. These variations are not the same when a change in power losses is generated in the power module as can be seen from Figures 4.16 and 4.17.

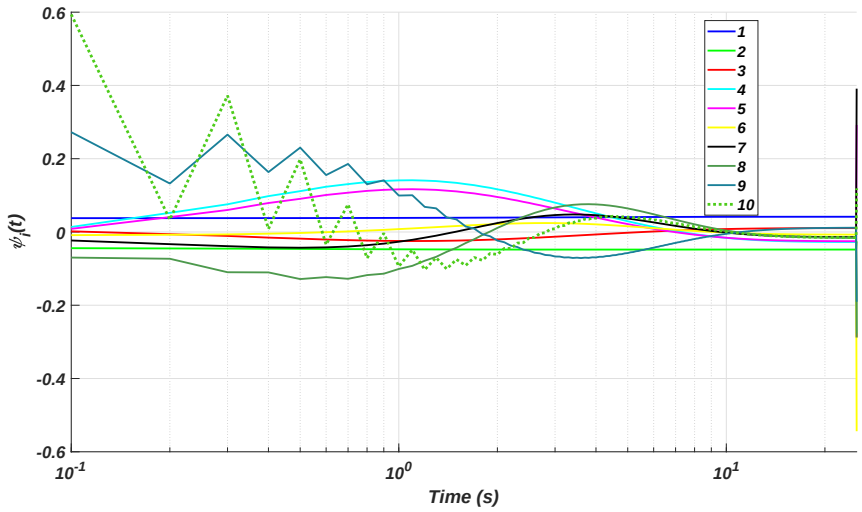


Figure 4.16 – The first 10 temporal modes for $0\ s \leq t \leq 25\ s$

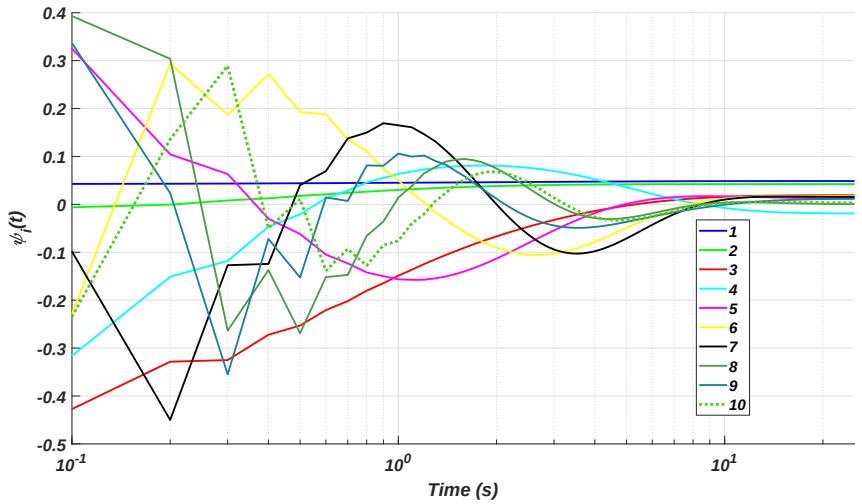
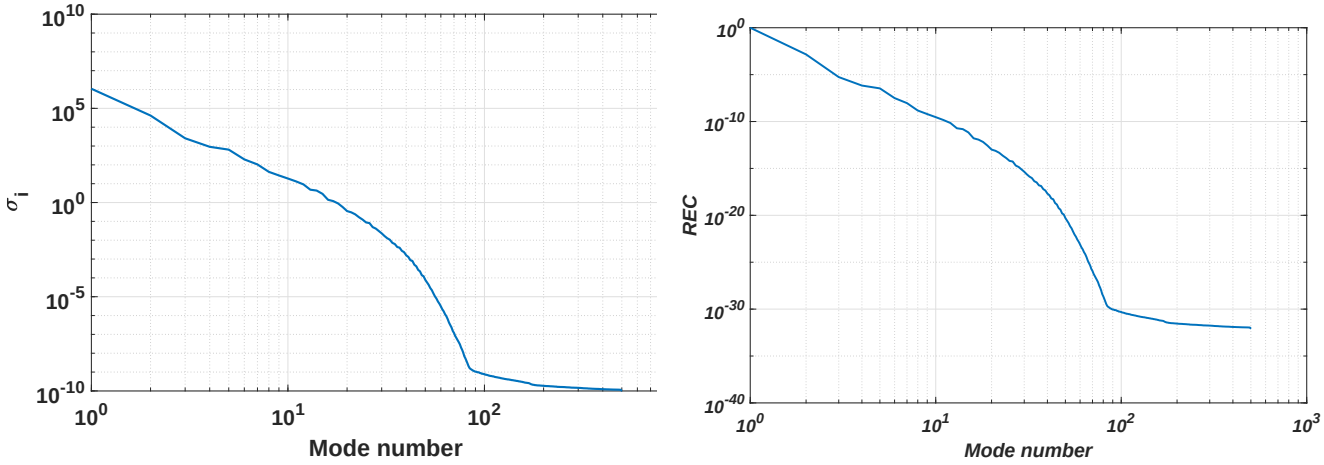


Figure 4.17 – The first 10 temporal modes for $25\ s < t \leq 50\ s$

The singular value distribution and relative energy content are presented on Figure 4.18.



Singular value distribution — Relative Energy Content (REC)

Figure 4.18 – Singular values and REC distribution

As explained in the first section, the reconstruction error using the r first modes is bounded by $\sqrt{min(M,N)-r}\ \sigma_{r+1}$ (see equation (3.37)). In order to realize a reconstruction having an error less or equal to $0.1\ ^{\circ}C$, the number of retained modes should be greater or equal to 25 modes.

If we want to keep modes having a relative energy constribution greater or equal to 10^{-10}, then the number of retained modes should be greater or equal to $r = 15$ modes.

A graphic illustration of the singular value decomposition of the temperature field is given in Figure 4.19. The SVD can be seen as the method of separation of variables. The spatial part are called the POD or spatial modes. The product singular value times the temporal mode is called the temporal amplitude. Each of the spatial and temporal modes are orthonormal(they are all unit vectors and orthogonal to each other). Therefore the weight of the snapshots matrix is localized in the singular values. The main feature of the singular value decompostion is the exponential decrease of its singular values as it can be seen from Figure 4.18. This feature allows to write each column of the snapshots matrix as a linear combination of a reduced number of vectors (dominant POD modes).

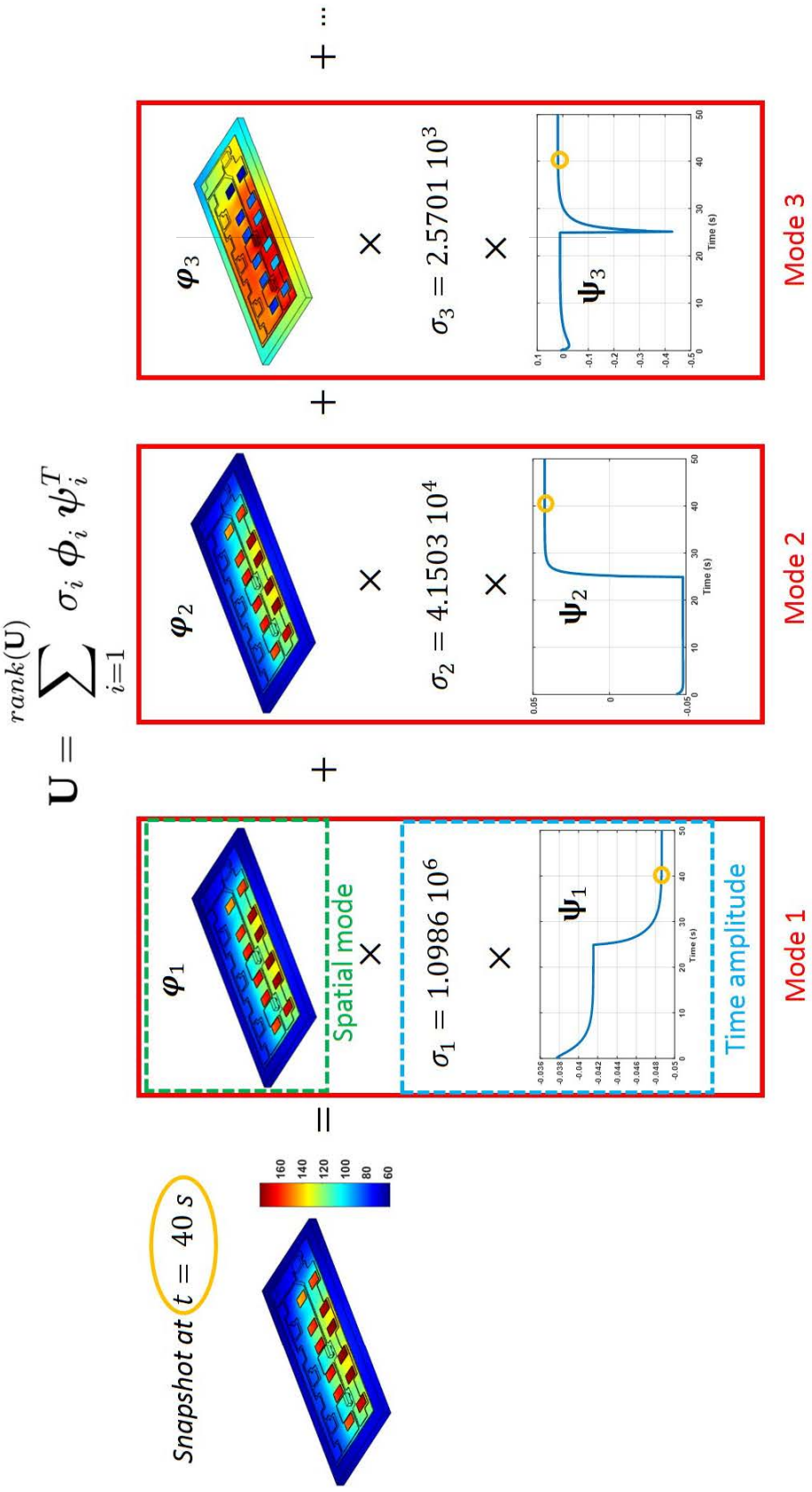


Figure 4.19 – Temperature field decomposition using SVD

In order to assess the dominant modes selection criteria based on singular value distribution, we plot the reconstruction error for each snapshot using r modes on Figure 4.20 and Figure 4.21.

Two recosntruction errors have been calculated:

The first one (Figure 4.20) is the l^2 norm of the difference between the baseline snapshot $\mathbf{u}(t_i)$ and the one reconstructed from the first r modes, namely:

$$\|\mathbf{u}(t_i) - \mathbf{u}_r(t_i)\|_2$$

The second one (Figure 4.21) is the maximum norm of the difference between the baseline snapshot $\mathbf{u}(t_i)$ and the one reconstructed using the first r modes, namely:

$$\|\mathbf{u}(t_i) - \mathbf{u}_r(t_i)\|_\infty$$

The second error can be seen as the maximum of the absolute error over the nodes.

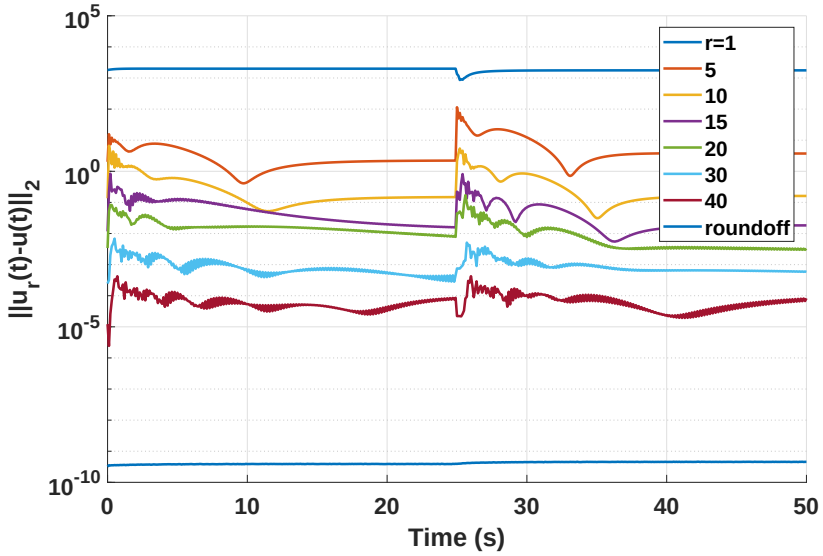


Figure 4.20 – Reconstruction error of the snapshots matrix using the l^2 norm

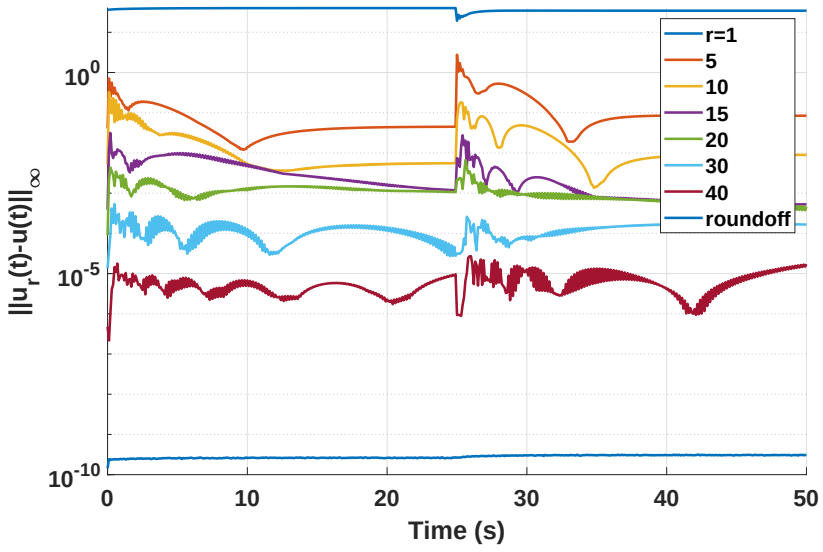


Figure 4.21 – Reconstruction error of the snapshots matrix using the maximum norm

Figure 4.22 shows that the reconstruction error raises during the sharp transient phases (sudden variation of the power losses or the cooling fluid temperature). This error falls when the number of retained modes increases. It can be noticed from Figure 4.22 that $r = 15$ modes are required to obtain an approximation of the temperature field with an error less or equal to $0.1\ ^{\circ}C$. The dominant modes selection criteria based on the singular value distribution states that $r = 25$ modes are required to have an approximation error less or equal to $0.1\ ^{\circ}C$ (see Figure 4.18). Therefore, the selection criteria based on the singular values distribution overestimates the number of retained modes. This can be explained by the fact that singular values give an estimation of the reconstruction error for the **total** matrix and not for each snapshot.

Now, the reconstruction error of the snapshots matrix is plotted as function of r. Two errors are plotted:

- The first error is the Frobenius norm of the difference between the baseline snapshots matrix and the one reconstructed from the first r modes. This error can be written as :

$$E(r) = \|\mathbf{U} - \mathbf{\Phi}(:, 1:r)\ \mathbf{\Sigma}(1:r, 1:r)\ \mathbf{\Psi}(:, 1:r)^T\|_F$$

- The second error is computed as the maximum in time and space of the difference between the baseline snapshots matrix and the one reconstructed from the first r modes, namely :

$$max_{time}(max_{\mathbf{x}}(|\mathbf{U} - \mathbf{U}_r|))$$

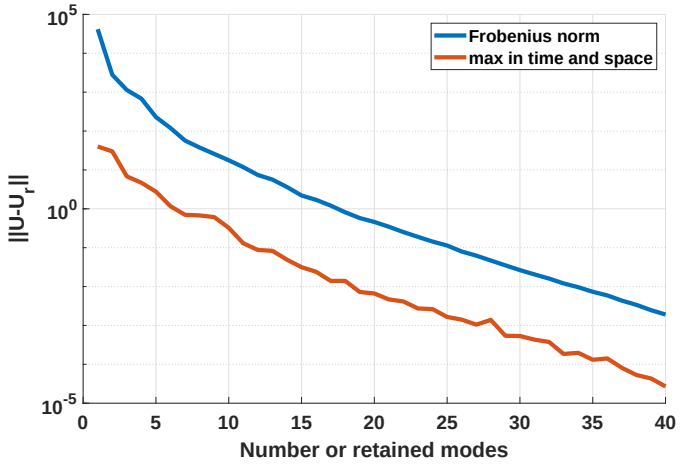


Figure 4.22 – Reconstruction error of the snapshots matrix

It can be noticed from Figure 4.22 that a ratio of 100 exists between the two types of errors. The second error is more relevant than the first one as it gives an exact value of the maximum reconstruction error.

Based on this statement, the number of retained modes that realizes a maximum error less than $0.1\ ^{\circ}C$ is $r = 13$ modes.

In the next section a reduced order model consisting of 10 modes will be used. The temperature field will be computed in the basis containing the first 10 POD modes.

4.6.2 Usage stage

The power profile used to construct the POD basis is presented on Figure 4.12. The issue that arises is: does the ROM remains accurate if the power profile vary with respect to the learning one?

We remind that in practical situations, the power profile vary constantly. In this section we investigate the robustness of the ROM when considering a power profile different from the learning one. The power profile considered in this study is presented on Figure 4.23.

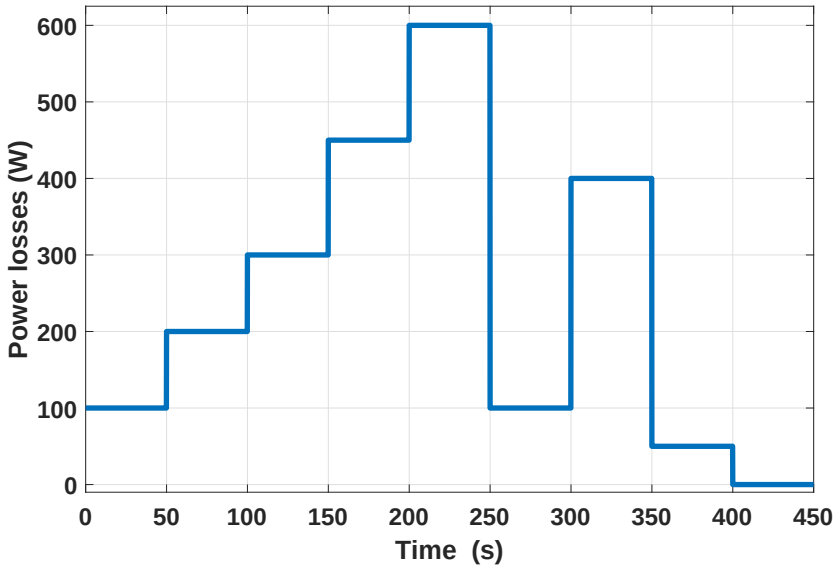


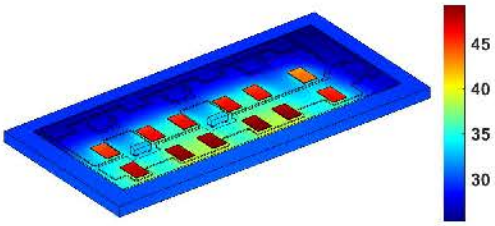
Figure 4.23 – Usage power profile

The boundary conditions (h_{eq} and T_f) are kept constant.

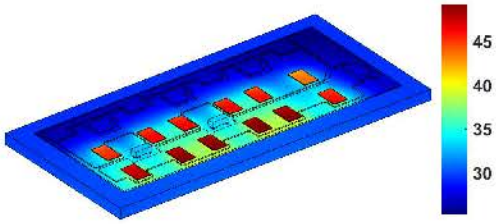
In this study the FEM and the ROM are simulated.

The surface temperature of the ROM and the FEM at different instants ($t = 1\ s$, $t = 55\ s$ and $t = 110\ s$) in the case of $r = 10$ retained modes are presented on Figure 4.24. A considerable resemblance between the ROM and the FEM can be seen on Figure 4.24 at each time step.

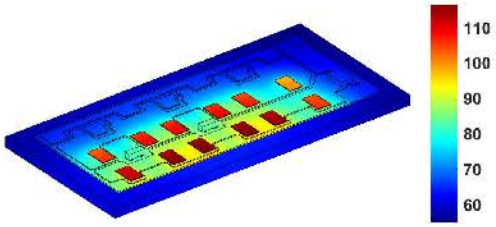
The absolute error between the FEM and the ROM at the same instants is plotted on Figure 4.25. There is no typical spatial behavior of the error over the power module. This error do not exceed $0.1\ °C$ for the presented instants. Obtained results show a good agreement between the ROM and the FEM.



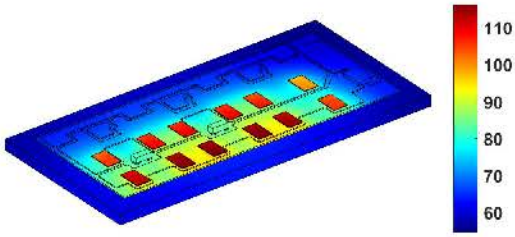
FEM at $t = 1\ s$



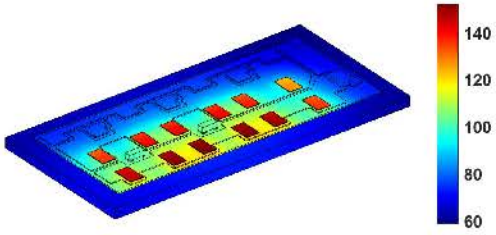
ROM at $t = 1\ s$



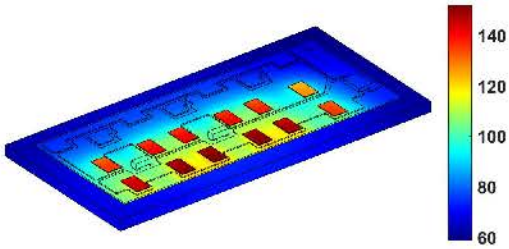
FEM at $t = 55\ s$



ROM at $t = 55\ s$

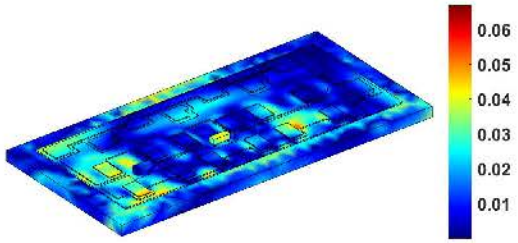


FEM at $t = 110\ s$

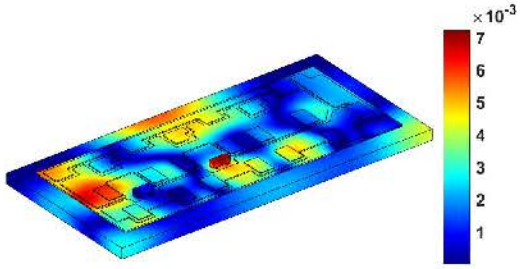


ROM at $t = 110\ s$

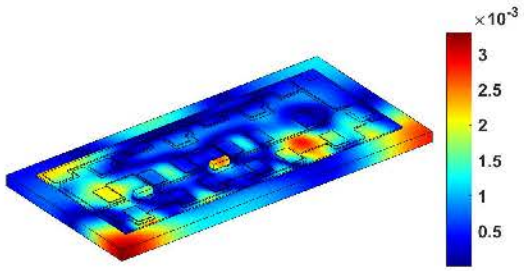
Figure 4.24 – The temperature field given by the FEM and the ROM at instants $t = 1\ s$, $t = 55\ s$ and $t = 110\ s$



$t = 1\ s$



$t = 55\ s$



$t = 110\ s$

Figure 4.25 – The absolute difference between the FEM and the ROM of order 10 at instants $t = 1\ s$, $t = 55\ s$ and $t = 110\ s$

The time evolution of temperature in the hottest node is plotted on Figure 4.26.

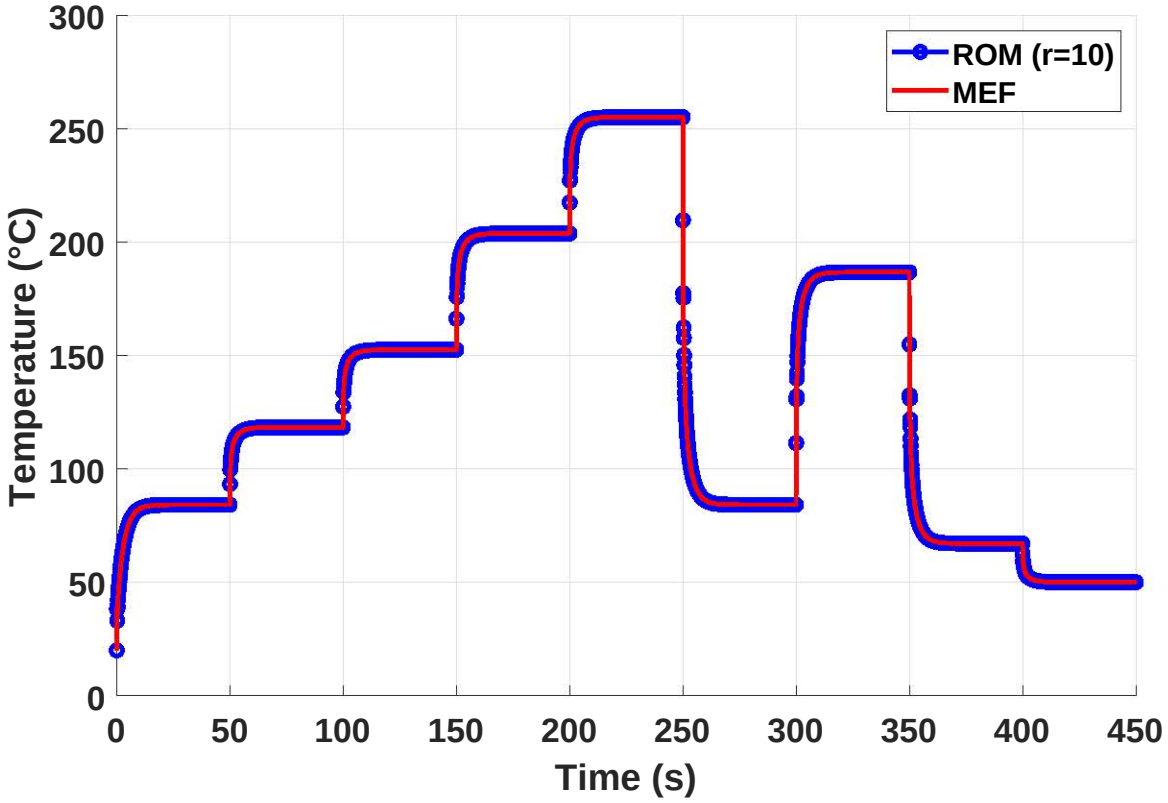


Figure 4.26 – Temperature profile in the hottest node given by the FEM and the ROM of order 10

A good matching between the FEM and the ROM can be observed. This agreement is maintained for large range of power losses.

In order to quantify the error of the ROM, the maximum norm of the difference between the FEM and the ROM is plotted as function of time on Figure 4.27. This error is plotted for several values of retained POD modes $r = 5, 10, 15, 20, 30$. It can be seen that the ROM of order 10 has an error that does not exceed $0.4\ °C$ over the time and space. Whatever the value of the power losses, the error remains low and there is no typical behavior of the ROM error with respect to power losses. It can be observed that increasing the number of retained modes improves the accuracy of the reduced order model. In practice, an error less than $0.1\ °C$ does not affect the lifetime of an electronic component.

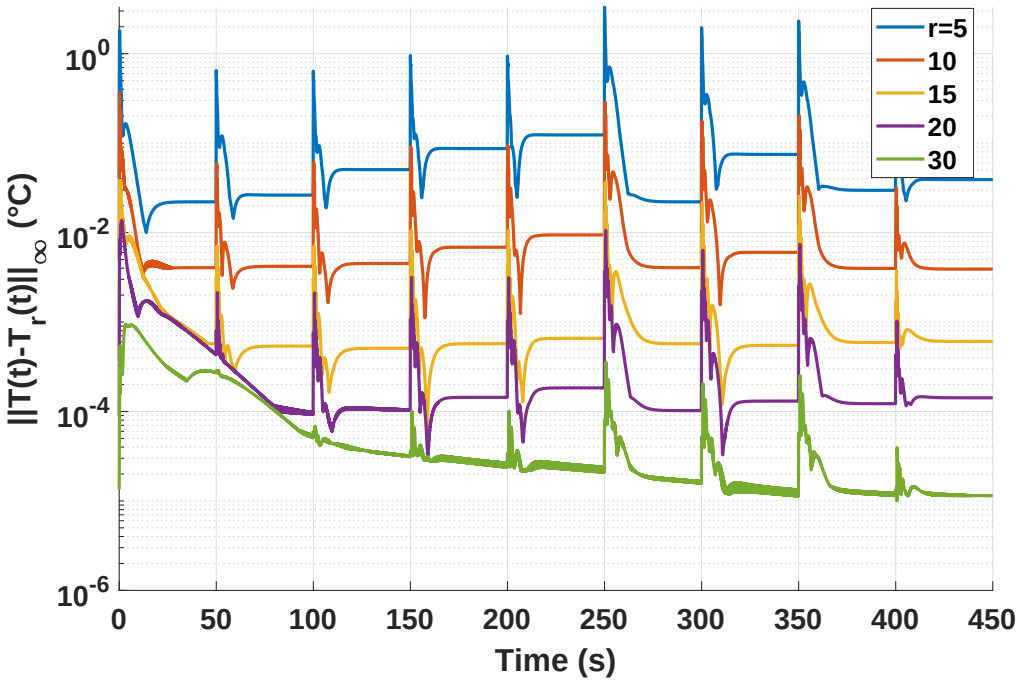


Figure 4.27 – Usage power profile

It has been proven in the previous results that the ROM of order 10 has an error that does not exceed $0.4\ °C$ for a large range of power losses. The question then arises is the computing time gain realized with such model. The computing time is evaluated for both FEM and ROM and is presented in Table 4.4.

Model	FEM	ROM (r=10)	ROM (r=20)	ROM (r=30)
CPU Time	11 hours	1.7 seconds	1.8 seconds	1.9 seconds
Gain		23 000	22 000	21 000

Table 4.4 – The computing time of the FEM and the ROM

It can be noticed that the CPU time is more or less the same between the ROM with $r = 10, 20, 30$. A relevant choice will be to use a ROM consisting of $r = 30$ modes.

4.6.3 ROM limits: time step effect

The finite element solution has been computed in the case of $\Delta t_{FEM} = 0.1\ s$ using Crank-Nicolson time integration. The singular value decomposition can be computed using all snapshots or a sub-sampling of the snapshots with a chosen time step that will be denoted Δt_{SVD}.

The reduced order model can be computed using a different time step from the two previous ones, that will be denoted by Δt_{ROM}.

All these time steps are illustrated on Figure 4.28.

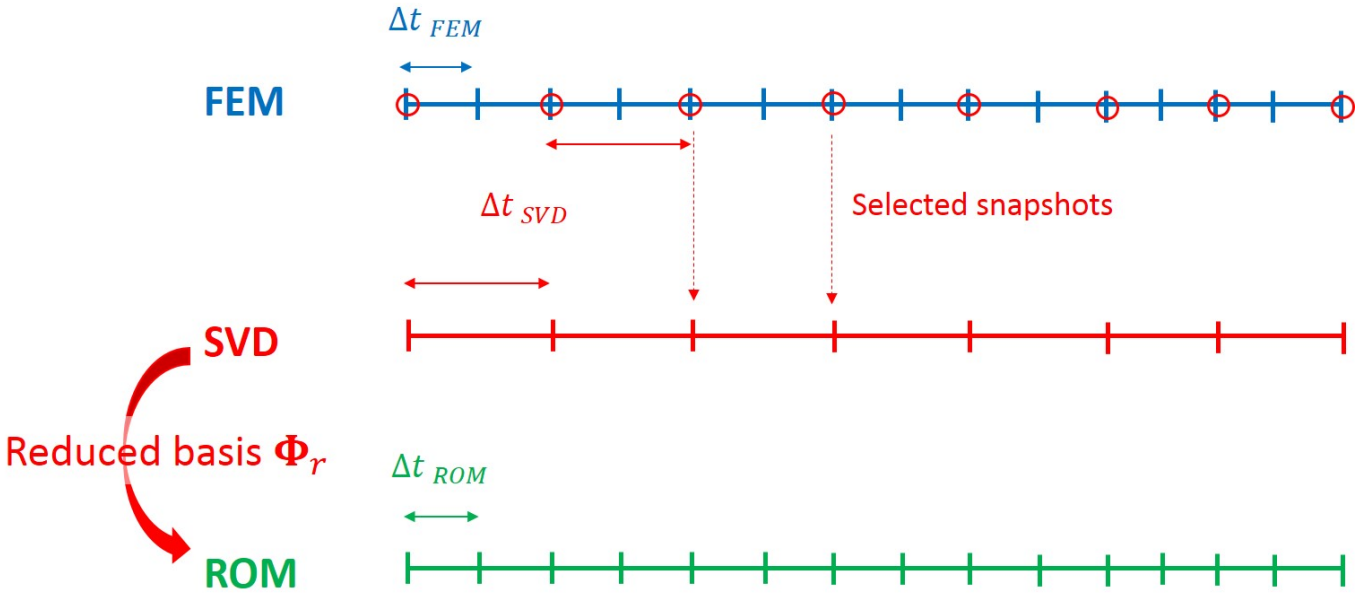


Figure 4.28 – The different time steps

In this section we will investigate the effect of Δt_{SVD} on the accuracy of the ROM.

For this purpose, we consider a set of snapshots time steps Δt_{SVD} respecting a power of 2 ratio, namely : $\Delta t_{SVD} = \Delta t_{FEM}$, $\Delta t_{SVD} = 2\Delta t_{FEM}$, $\Delta t_{SVD} = 2^2\ \Delta t_{FEM}$, $\Delta t_{SVD} = 2^3\ \Delta t_{FEM}$, $\Delta t_{SVD} = 2^4\ \Delta t_{FEM}$.

The ROM is executed with the same time step as the FEM: $\Delta t_{ROM} = \Delta t_{FEM}$, but with a POD basis obtained with different Δt_{SVD} time steps. The considered ROM consists of 10 dominant modes. The maximum norm error is presented on Figure 4.29.

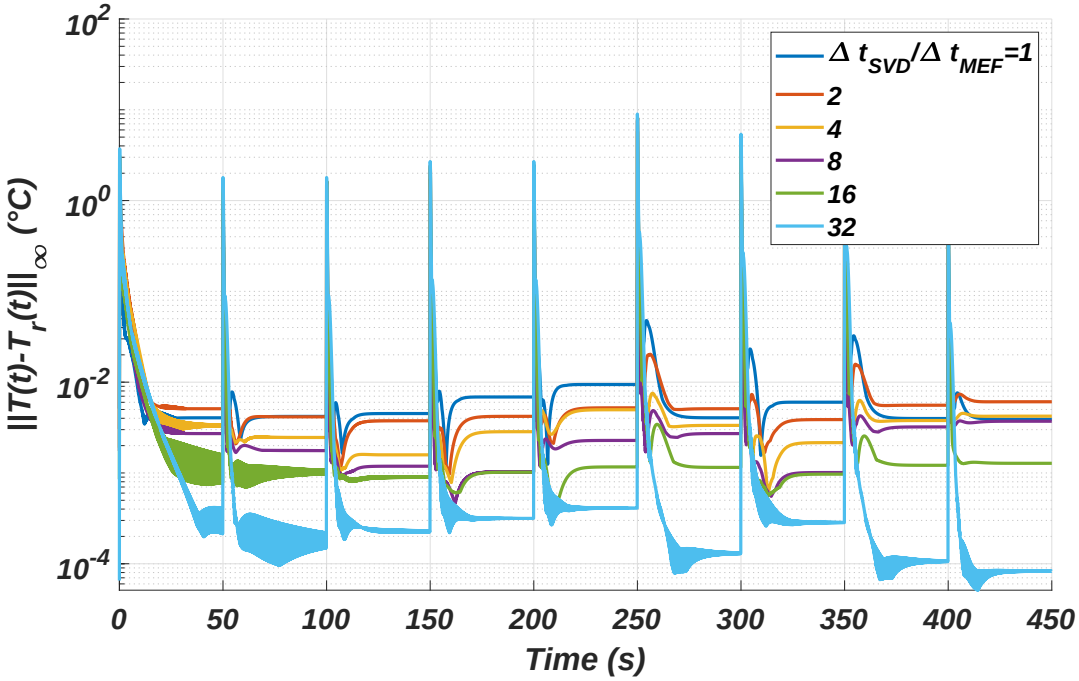


Figure 4.29 – ROM (r=10) in the case of different Δt_{SVD}

Figure 4.29 shows that raising Δt_{SVD} improves the accuracy at the steady state.

During the suddent variation of the power profile, the error is important for all Δt_{SVD}.

In order to better visualize the effect of Δt_{SVD} on the accuracy of the ROM, only the first 50 s have been considered and the time-axis is set to a logarithmic scale .

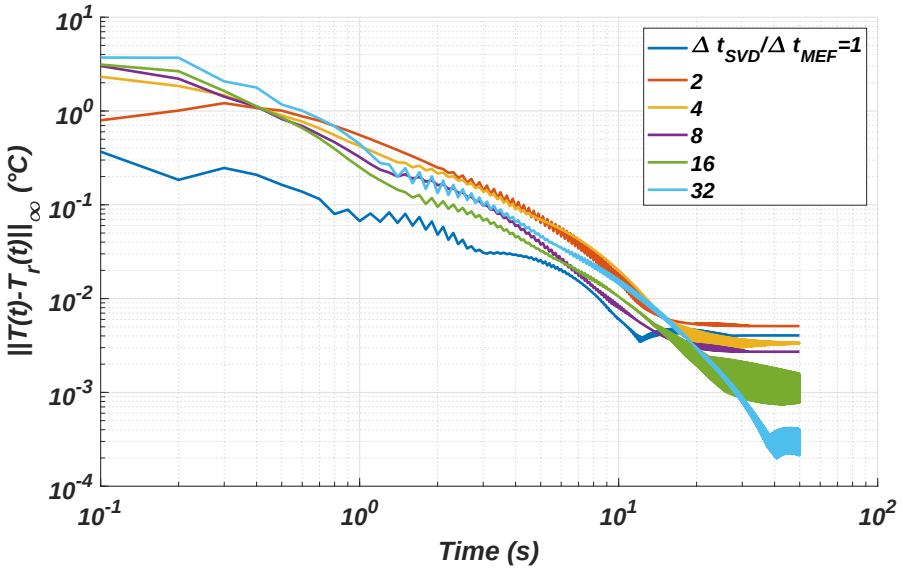


Figure 4.30 – ROM (r=10) in the case of different Δt_{SVD}

It can be seen from Figure 4.30 that during the first $15\ s$, the ROM obtained with $\Delta t_{SVD} = \Delta t_{MEF}$ has the better accuracy. Increasing the snapshots time step Δt_{SVD} decreases the accuracy of the ROM during the transient phase.

It is outlined in section 4.6.2 that the error raises up when changing power losses and that increasing the number of retained modes decreases this error. In the Figure 4.31, the sudden variation time region (around $50\ s$) is zoomed in. It can be stated that increasing the time step Δt_{SVD} deteriorates the accuracy of the ROM during this period.

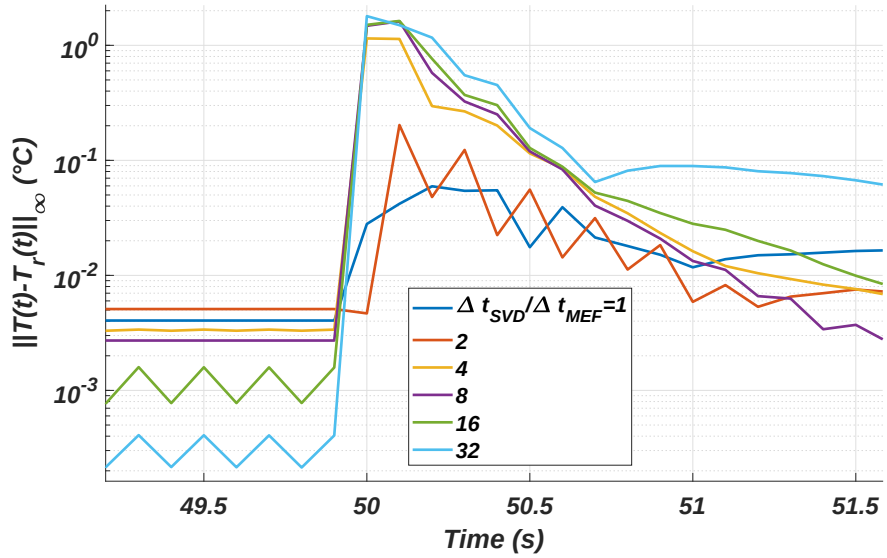


Figure 4.31 – Zoom around $t = 50\ s$ of the ROM(r=10) error in the case of different Δt_{SVD}

Two important conclusions can be drawn from the results :

- Increasing the snapshots time step Δt_{SVD} enhances the accuracy of the ROM during steady state period.
- Decreasing the snapshots time step Δt_{SVD} enhances the accuracy of the ROM during transient state period.

The question is : can we predict such behavior during the learning stage before performing ROM simulation ?

The answer is YES and it will be by assessing the projection error of the baseline snapshots matrix (computed with Δt_{MEF}).
The procedure steps are described below :

1. Constructing the snapshots matrix with a sub-sampling time step Δt_{SVD} from the baseline snapshots matrix .
2. Constructing a POD basis from the snapshots matrix obtained by a time subsampling time step Δt_{SVD}. In the POD basis only $r = 10$ are kept. This POD basis will be denoted $\phi_{r,\Delta t_{SVD}}$.
3. Projecting the baseline snapshots matrix on the POD basis obtained by Δt_{SVD}. The projection is computed by the following relationship :

$$\left(\phi_{r,\Delta t_{SVD}}.\phi^T_{r,\Delta t_{SVD}}\right).\mathbf{U}$$

The time evolution of the projection error is computed as follows:

$$E_{projection}(r, \Delta t_{SVD}, t_i) = ||u(t_i) - \left(\phi_{r,\Delta t_{SVD}} . \phi^T_{r,\Delta t_{SVD}}\right) . u(t_i)||_\infty$$

This error is presented on Figure 4.33.

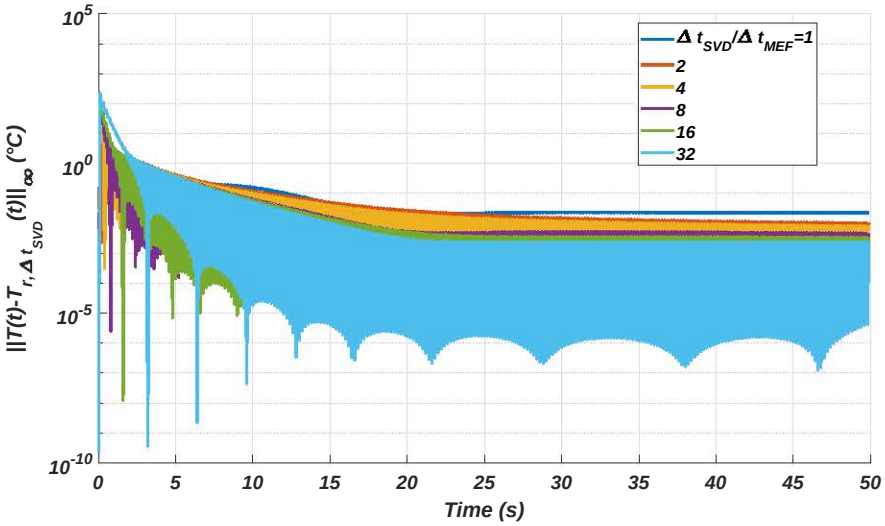


Figure 4.32 – Projection error of the snapshots matrix in the case of different Δt_{SVD}

In order to investigate the error during a short time period, the current time-axis is set to a logarithmic scale.

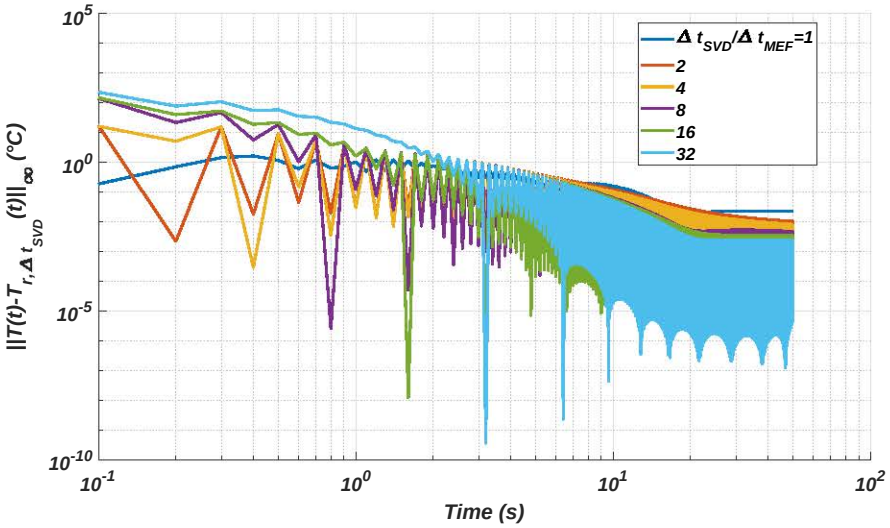


Figure 4.33 – Projection error of the snapshots matrix in the case of different Δt_{SVD}

It can be seen from Figure 4.33 that the error behavior is roughly similar to the one presented on Figure 4.30.

Therefore, it is recommended to compute the projection error for different Δt_{SVD}, if we intend to use a time step different from learning.

4.7 Conclusions

Two reduced order models have been constructed for two purposes: cooling system design and temperature monitoring within the power module. The performances of both ROM is assessed in terms of accuracy and computing time.

The constructed steady ROM consists of 4 POD modes. This model provides accurate results with an error less or equal to $0.03\,°C$. This accuracy is maintained for large range of power losses and heat coefficient. The steady ROM allows to divide the computing time by 267.

The transient ROM consists of 10 modes. The selection criteria of dominant POD modes have been justified. A good agreement has been observed between the 3-dimensional temperature field given by the FEM and the ROM. The time evolution of the error shows that the accuracy is degraded when power losses vary suddenly and that increasing the number of retained modes reduces these error peaks. It has been shown that the accuracy of the transient ROM is preserved for large range of power losses. We have noted that increasing the time step between snapshots declines the accuracy of the ROM accuracy during the transient state and improves it during steady state. For the usage power profile, the computing time gain is equal to 23000.

In conclusion, the reduced order models constructed using POD and Galerkin projection has shown to be accurate and fast to study heat transfer in electronic components.

Bibliography

[1] Ho-Ming Tong, Yi-Shao Lai, and CP Wong. *Advanced flip chip packaging*. Springer, 2013.

[2] Cheikh Tidiane Dia. *Génération de modeles compacts thermiques dynamiques de composants electroniques via les algorithmes genetiques*. PhD thesis, Paris 10, 2015.

[3] PH Mellor, D Roberts, and DR Turner. Lumped parameter thermal model for electrical machines of tefc design. In *IEE Proceedings B (Electric Power Applications)*, volume 138, pages 205–218. IET, 1991.

[4] Ayman M El-Refaie, Nathan C Harris, Thomas M Jahns, and Khwaja M Rahman. Thermal analysis of multibarrier interior pm synchronous machine using lumped parameter model. *IEEE Transactions on Energy Conversion*, 19(2):303–309, 2004.

[5] Xinfan Lin, Hector E Perez, Shankar Mohan, Jason B Siegel, Anna G Stefanopoulou, Yi Ding, and Matthew P Castanier. A lumped-parameter electro-thermal model for cylindrical batteries. *Journal of Power Sources*, 257:1–11, 2014.

[6] Shafigh Nategh, Oskar Wallmark, Mats Leksell, and Shuang Zhao. Thermal analysis of a pmasrm using partial fea and lumped parameter modeling. *IEEE Transactions on Energy Conversion*, 27(2):477–488, 2012.

[7] Christophe Forgez, Dinh Vinh Do, Guy Friedrich, Mathieu Morcrette, and Charles Delacourt. Thermal modeling of a cylindrical lifepo4/graphite lithium-ion battery. *Journal of Power Sources*, 195(9):2961–2968, 2010.

[8] Masaru Ishizuka, Tomoyuki Hatakeyama, Yuichi Funawatashi, et al. Application of thermal network model to transient thermal analysis of power electronic package substrate. *Active and Passive Electronic Components*, 2011, 2011.

[9] Francesc N Masana. A closed form solution of junction to substrate thermal resistance in semiconductor chips. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A*, 19(4):539–545, 1996.

[10] SM Thompson and HB Ma. Thermal spreading analysis of rectangular heat spreader. *Journal of Heat Transfer*, 136(6):064503, 2014.

[11] FN Masana. A straightforward analytical method for extraction of semiconductor device transient thermal parameters. *Microelectronics Reliability*, 47(12):2122–2128, 2007.

[12] Angelo Raciti, Davide Cristaldi, Giuseppe Greco, Giovanni Vinci, and Gaetano Bazzano. Electrothermal pspice modeling and simulation of power modules. *IEEE Transactions on Industrial Electronics*, 62(10):6260–6271, 2015.

[13] M Michael Yovanovich. Four decades of research on thermal contact, gap, and joint resistance in microelectronics. *IEEE transactions on components and packaging technologies*, 28(2):182–206, 2005.

Chapter 5

Inverse reduced order model of the three legs power electronic module

Sommaire

5.1 Introduction

Problem statement

The performance of semiconductors devices such as MOSFET transistors depends deeply on temperature [1]. In order to predict failure that can occur in electronic components and pursue their aging, the temperature in the power module should be monitored in real time.

Placing a sensor on each transistor is prohibited because of packaging constraints. To bypass this problems, temperature measurement models have been developed. These models can be either direct or inverse. Direct models require the knowledge of power losses in passive components and boundary conditions, whereas inverse models can work with any physical quantities of the thermal problem, for instance the temperature in allowed measurement locations.

State of the art

Several direct models are used to compute the temperature of electronic components such as thermal network models. These models consists of a ladder of thermal resistances and capacitances that are calculated either from physical and geometric properties (Cauer model) [2] or identified by a minimization algorithm (Foster model) [3,4]. Numerical models based on space discretization such as finite element method are also used to compute the temperature field in electronics. The main drawback of this class of methods is its high computing time, thus they can not be used to monitor in real time the temperature. Reduction methods such as POD are used to decrease the dimension of the detailed model while conserving its accuracy. These models show good performance when treating industrial cases, for instance temperature calculation of a buried oil pipeline [5], two-dimensional temperature field in a gas-cooled turbine blade [6] and 3D temperature field in a power electronic module [7].

All the previous models are direct, but in many practical situations, it is difficult to have access to power losses or boundary conditions. For instance in high power electronic modules, the strong electro-magnetic waves will disturbe the current measurement and the on-state electrical resistances of MOSFETs are not known with precision, thus the obtained power losses are inaccurate. To overcome this obstacle, inverse techniques are used and aim to measure the temperature in allowed locations in order to construct the missing information in the zone of interest. For example, S.R.Carvalho et al [8] constructed a numerical thermal model for the cutting tool using finite volume method. Then they minimized the square of the difference between the measured temperature and the temperature obtained by thermal model in order to identify the heat flux at the chip–tool interface.

Computing the detailed model inside the inverse model loop leads to high computing time. To bypass this limit, some authors used the reduced model. For example M.Girault et al [9, 10] used a reduced order model based on the model identification method (MIM) to solve the inverse heat conduction problem (IHCP). They used the future time steps method in order to estimate a time dependent thermal input from the temperature measurement inside the domain. E.Videcoq et al [11] used the same approach based on branch eigen modes reduced order model. The model was shown efficient to compute the temperature of the cutting tool in the friction zone which is inaccessible by direct measure of temperature.

The proposed solution

In the last approaches, we must first identify the power losses and then compute the temperature in inaccessible locations.

The inverse reduced order model (IROM) skips the stage of power losses identification and establishes a direct relation between the measured temperature and the temperature in inaccessible locations.

In this chapter an illustration of the IROM in the case of the three-legs power electronic module is given. In sections 2, the geometry and the materials of the power module as well as the boundary and initial conditions are described. In section 3, a brief description of the used finite element model is given.

In sections 4 and 5, the construction (learning stage) of the IROM and its accuracy assessment (usage stage) are depicted. The effect of the locations and number of measurement are adressed.

5.2 The studied power electronic module (PEM)

5.2.1 Geometry and materials of the PEM

The studied power module is presented in Figure 5.1. This system consists of 3 legs (phases) power modules mounted on the cooling system. The geometry and material description of a single leg power module is given in Chapter 4. The spacing between two consecutive legs is $5\ mm$.

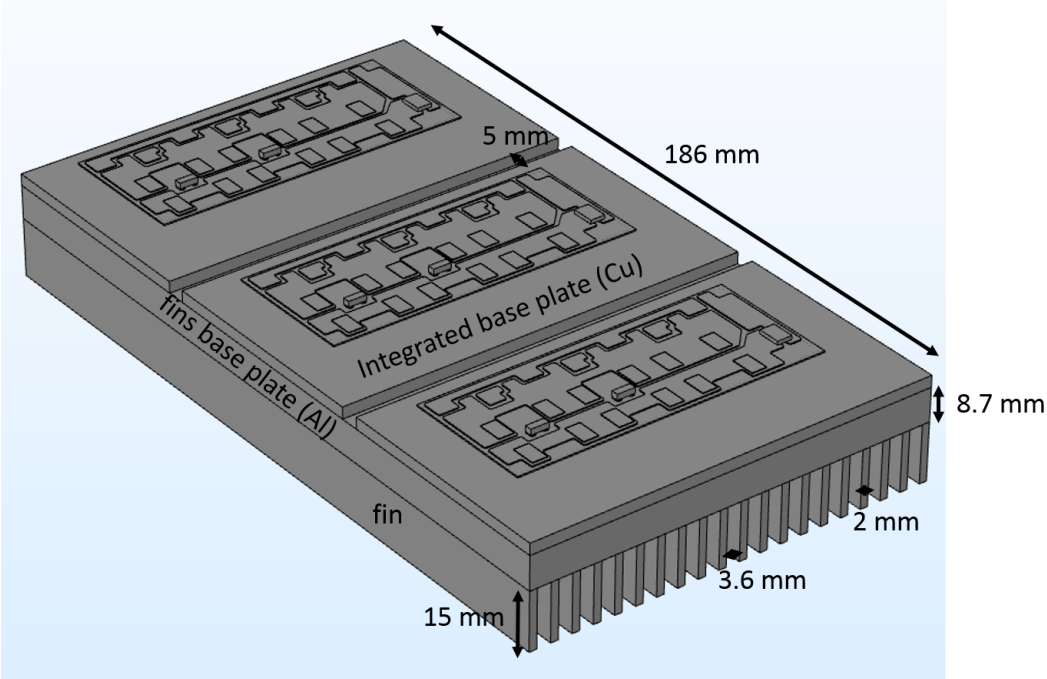


Figure 5.1 – The 3 legs power electronic module

The chosen material for the cooling system is aluminum thanks to its good thermal performance and its lightweight. The thermal physical properties of the aluminium used in this study are: $\rho_{Al} = 2700\ kg/m^3$, $k_{Al} = 238\ W/(m.K)$ and $C_{p,Al} = 900\ J/(kg.K)$.

The used fins are rectangular and their dimensions are presented in Figure 5.1. The cooling fluid that will cross the fins is air. The thermo-physical properties of air used in this study are : $\rho_{air} = 1.2\ kg/m^3$, $k_{air} = 0.025\ W/(m.K)$, $C_{p,air} = 1005.4\ J/(kg.K)$ and $\nu_{air} = 1.5\ 10^{-5}\ m^2/s$.

5.2.2 Electronic components and temperature sensors

The power module consists of 36 transistors and 6 temperature sensors. The transistors are MOSFETs. The used temperature sensors are thermistor of the negative temperature coefficient (NTC) type. The number and placement of sensors was fixed by the industrial partners taking into consideration electrical constraints. Locations of MOSFETs and temperature sensors are presented in Figure 5.2.

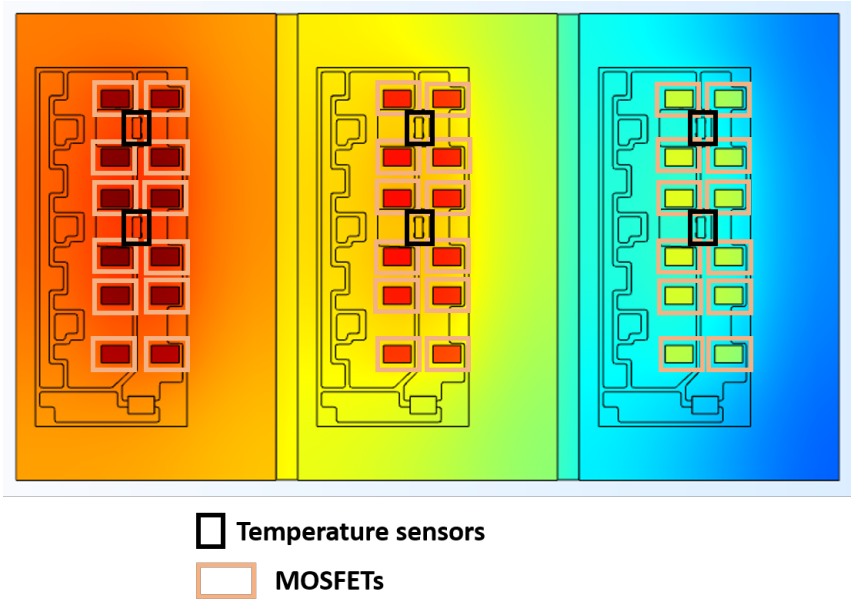


Figure 5.2 – MOSFETs and temperature sensors in the power module

5.2.3 Boundary and initial conditions

The fluid region boundary conditions are:

- The inlet air velocity denoted by $V_{f,in}$ is applied on the heat sink entrance.
- The relative pressure at the heat sink outlet is set to zero.

The heat transfer boundary conditions are:

- The inlet air temperature denoted by $T_{f,in}$ is set at the heat sink inlet.
- An adiabatic boundary condition is considered on the outer surface of the entire power module except the fluid inlet and outlet.

Power losses are assumed to be uniformly distributed over the 36 MOSFETs. The overall power losses is denoted by $P(t)$.

The temperature at $t = 0\ s$ is considered uniform over the solid and fluid regions and is equal to $T_{ini} = 20\ °C$.

5.3 The Finite Elements Model

The first step consists in performing a steady state fluid dynamic simulation of the fluid region. The flow is assumed laminar because the operating inlet velocity ranges between $2\ m/s$ and $8\,m/s$ (The Reynolds number ranges between 417 and 1668 which is less than 2400). During this step, the velocity and the pressure field are computed by solving the Navier-Stokes equations.

The second step consists in solving the energy equation in the fluid and in the solid regions. The computed velocity field is used in the energy equation for the fluid region.

These two steps have been implemented in the commercial finite element software $COMSOL^{®}$.

The used mesh is presented in Figure 5.3. Linear shape functions have been used in this study.

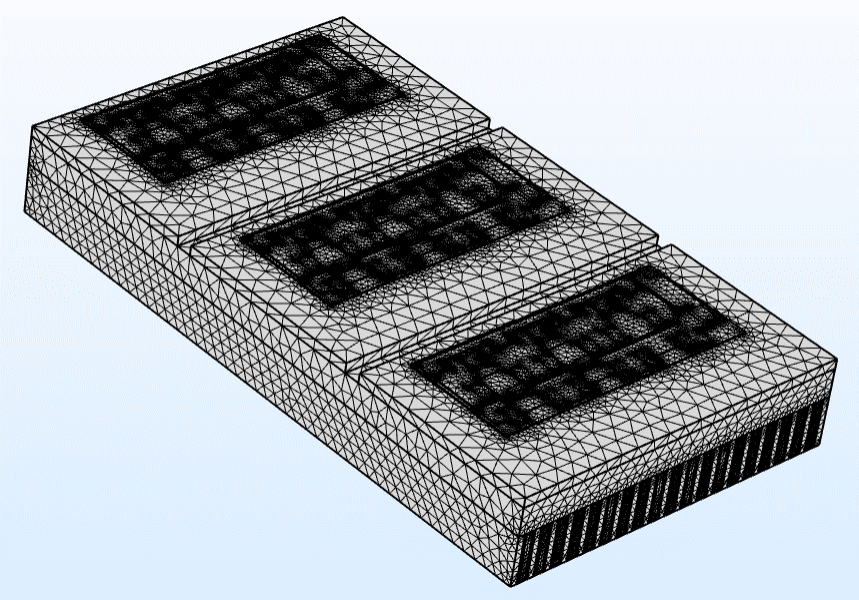


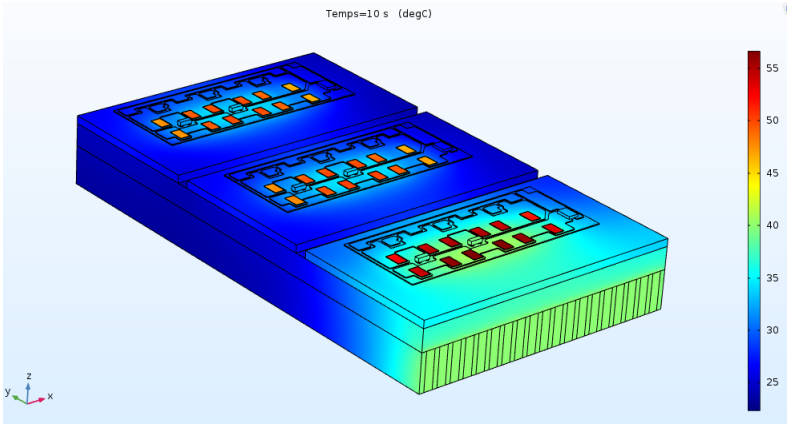
Figure 5.3 – The mesh

The mesh convergence is assessed by performing an extremely fine mesh simulation. The extremely fine mesh contains $3\,456\,182$ elements in the fluid region and $7\,737\,384$ elements in the solid region whereas the considered mesh contains $312\,074$ elements in the fluid region and $283\,366$ elements in the solid region. The relative difference of the maximum temperature between the two meshes is equal to $0.4\ \%$.

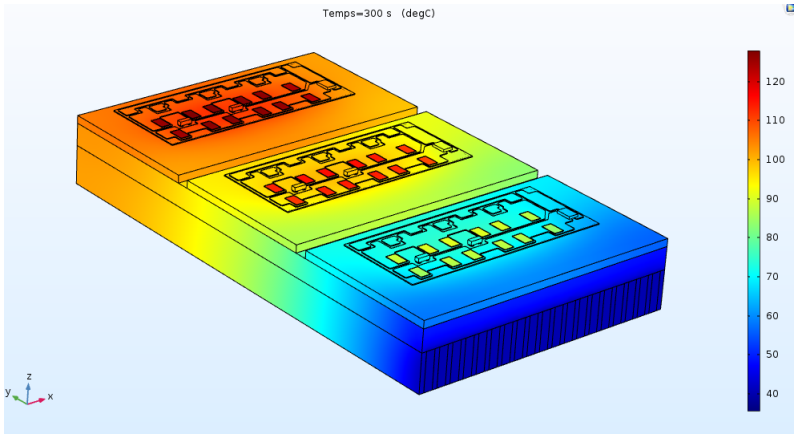
The numerical time integration of the finite element model is performed using a second order backward Euler scheme.

The first step to construct the inverse reduced order model is to compute the temperature field using the FEM. For instance, the temperature field at $t = 10\ s$ and $t = 300\ s$ are presented in Figure 5.4 in the case where $P(t) = 450\,W$ over the $300\,s$, $V_{f,in} = 5\,m/s$ and $T_{f,in} = 40\ °C$.

It can be seen in Figure 5.4 that the temperature along the fins (in the y-direction) is not uniform. This can be explained by the fact that the cooling air heats up when crossing the fins and removing the heat from electronic components.



$t = 10\ s$



$t = 300\ s$

Figure 5.4 – The temperature field in the power module at $t = 10\ s$ and $t = 300\ s$

5.4 Construction and assessment of the inverse reduced order model

5.4.1 Construction

1. **Construction of the finite element model (FEM) on $COMSOL^{®}$:**

   - Constructing of the power module geometry.
   - Assigning the material for each component of the geometry.
   - Setting the boundary conditions.
   - Constructing the mesh: in this task many meshes are assessed and the optimal one is kept.

2. **Construction of the IROM matrice A:**

- Recovering the finite element model results using LiveLink™ for $MATLAB^{®}$. The results considered in our study are the maximum temperature for each MOSFET and the mean temperature of each NTC sensor. This procedure yields the snapshots matrix.
- Performing a SVD on the snapshots matrix in order to extract a reduced basis ($\mathbf{\Phi}_r$) that represents well the temperature field behavior.
- Constructing the selection matrices $\mathcal{S}_{kn}$ and $\mathcal{S}_{unkn}$.
- Constructing the matrix **A** given by equation (3.70). This matrix links the MOSFETs temperature to the NTC temperature as shown in equation(3.71).

5.4.2 Assessment

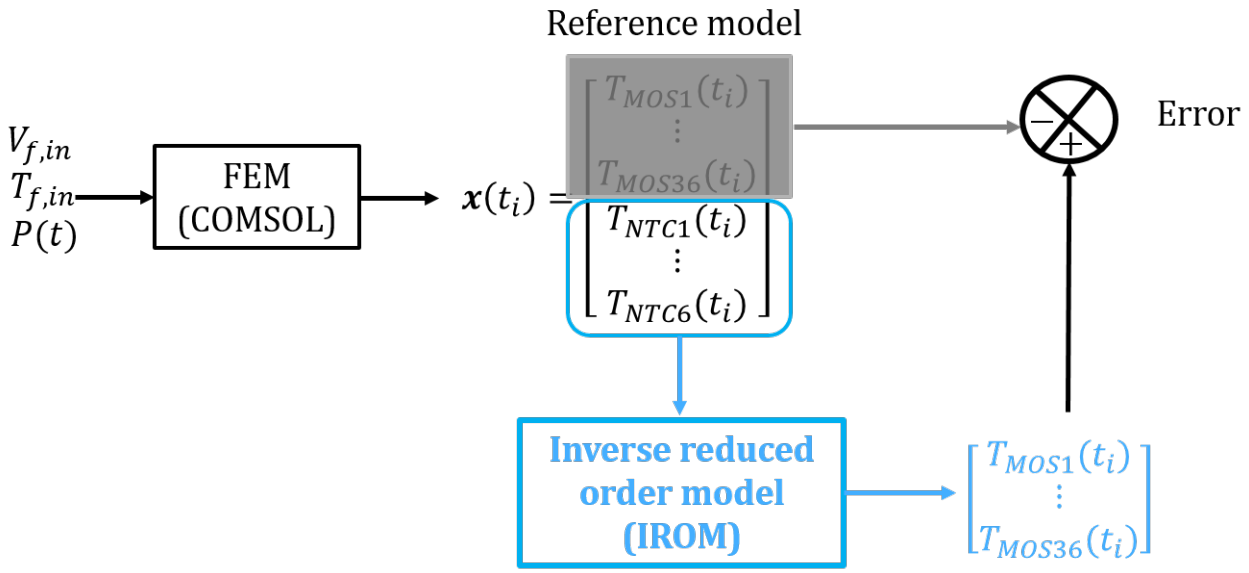


Figure 5.5 – Assessment of the inverse reduced order model

In order to assess the IROM, we run finite element simulation for given operating conditions (that can be different or identical to the learning). Then the maximum temperature of each MOSFET and the mean temperature of each NTC sensor is extracted. The MOSFETs temperature are supposed unknown and the NTC temperature are the input parameters for the IROM. The IROM calculation is launched. Obtained results are compared to the MOSFETs temperature given by the reference model. A schematic representation of this procedure is given in Figure 5.5.

5.5 Inverse Reduced Order Model of the Power Electronic Module

5.5.1 Learning stage: Spatial modes and eigen values distribution

In this stage, the extraction of the maximum temperature for each MOSFET and the mean temperature for each sensor is performed. As the power module consists of 36 MOSFETs and 6

temperature sensors, the snapshots matrix contains 42 rows. The time step between two consecutive snapshots is $\Delta t_{SVD} = 0.1\ s$ and the time simulation interval is $300\ s$. Therfore, the size of the snapshots matrix is $\boldsymbol{R}^{42\times 3001}$. The operating conditions used to construct the snapshots matrix are $P(t) = 450\,W$over the $300\,s$, $V_{f,in} = 5\,m/s$ and $T_{f,in} = 40\ °C$.

SVD is performed on the snapshots matrix. Obtained singular values are presented in Figure 5.6.

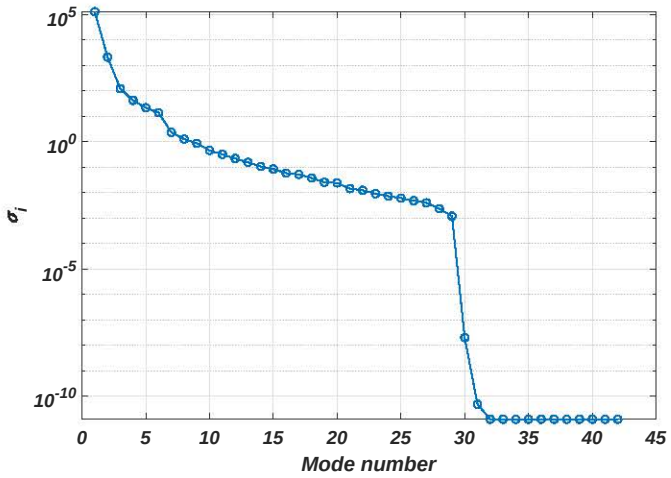


Figure 5.6 – Singular values distribution

Figure 5.6 implies that 15 modes are required to reconstruct the solution with a maximum error bounded by $\sqrt{42-15}\,\sigma_{16} = 0.26\ °C$ (see equation (3.37)).

The Relative Energy Content is presented in Figure 5.7.

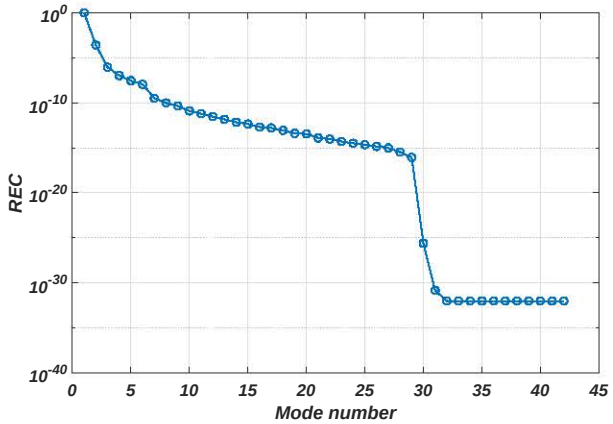


Figure 5.7 – Relative Energy Content (REC) distribution

Figures 5.7 implies that 5 modes should be retained in order to keep a relative information content greater or equal to 10^{-8}.

At present, only $r = 5$ modes are retained in the usage stage. The effect of retained modes on the accuracy of the model will be investigated in the following section. The matrix **A** introduced in Chapter 3 is computed from the $r = 5$ first POD modes.

5.5.2 Usage stage

5.5.2.1 Identical operating conditions with respect to learning

In this section, the same operating conditions as in the learning stage are considered. We assume that only the sensors temperature are known. The aim of the IROM is to compute the MOSFETs temperature, by knowing **only** the sensors temperature. The temperature in the hottest MOSFET is presented on Figure 5.8 for both the FEM and the IROM.

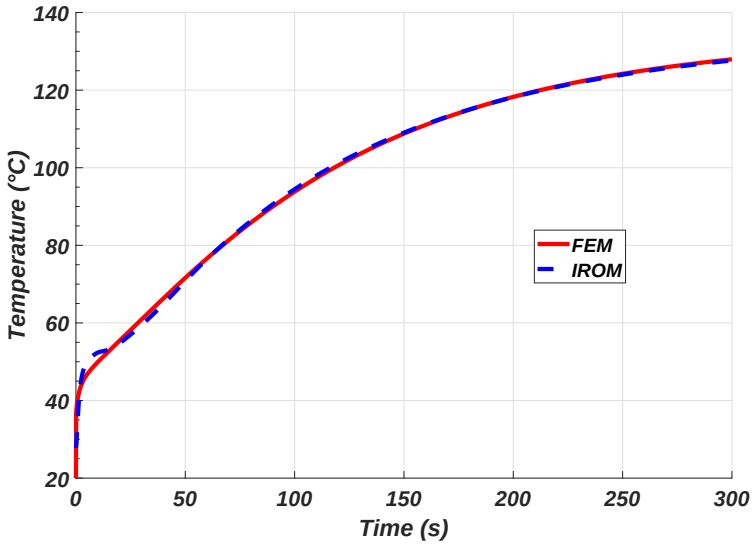


Figure 5.8 – Temperature profile in the hottest MOSFET

It can be seen from Figure 5.8 that a good agreement exists between the FEM and the IROM of order $r = 5$. A slight difference between the FEM and the IROM appears around $t = 5\,s$.

The maximum error for all MOSFETs is plotted as a function of time in Figure 5.9.

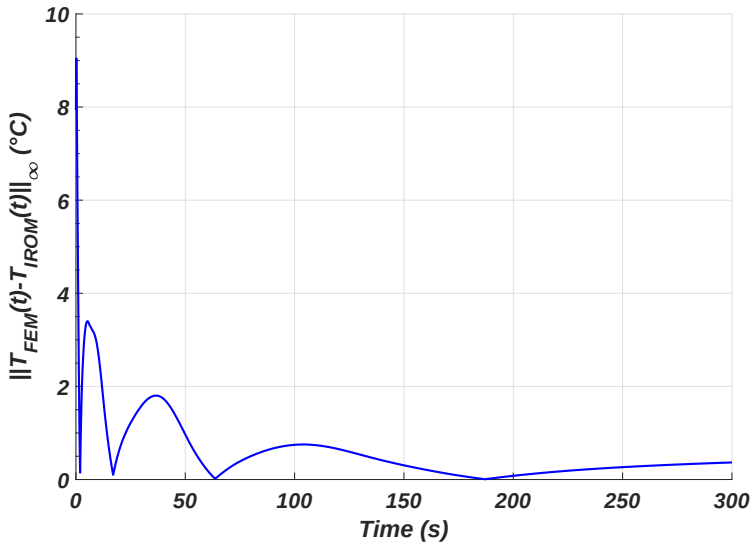


Figure 5.9 – The maximum error of the IROM

It can be seen that the error is important during the transient period. Beyond the first $50\,s$ the maximum error becomes less or equal to $1\,°C$

5.5.2.2 Effect of the number of retained modes on the accuracy of the IROM

In order to assess the robustness of the dominant modes selection for the IROM, the maximum error as function of time will be plotted for various values of retained modes $r = 1, \ldots, 19$.

In order to ensure a proper representation, the results are divided in two groups. We plot on Figure 5.10 the maximum error for $r \leq 6$ and on Figure 5.11 the maximum error for $r > 6$.

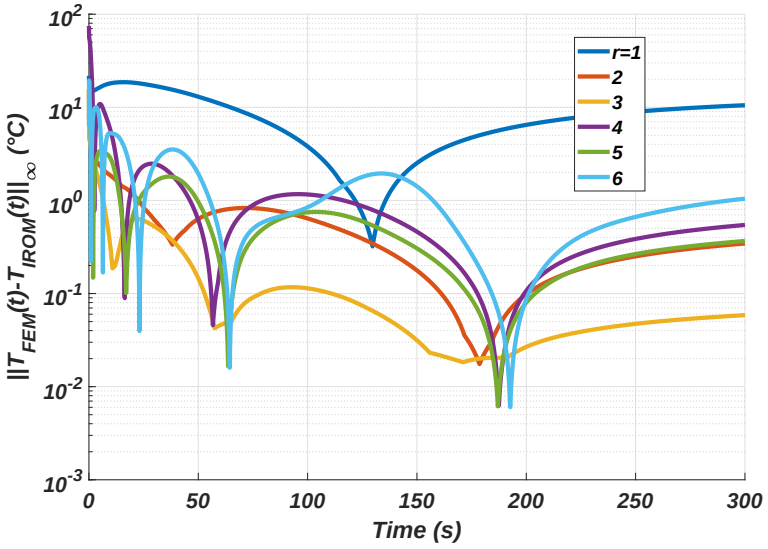


Figure 5.10 – The maximum error of the IROM for various values of retained modes $r = 1, \ldots, 6$

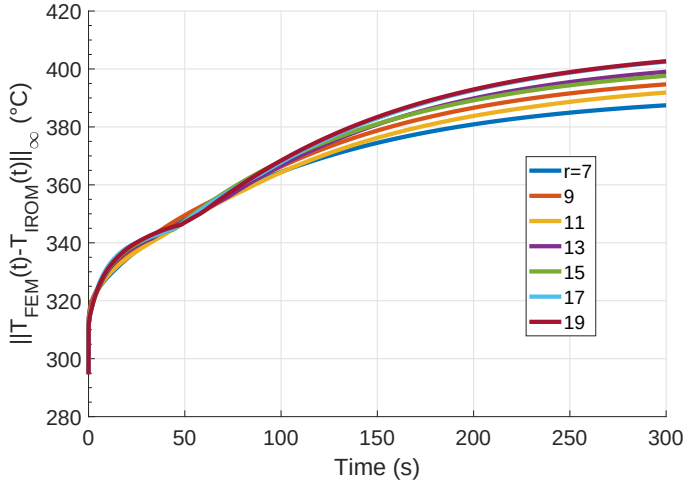


Figure 5.11 – The maximum error of the IROM for various values of retained modes $r = 7, \ldots, 19$

It can be seen that the error increases drastically when the number of retained modes becomes superior to the number of sensors $(r > 6)$. Such behavior is not expected because the

approximation of the temperature field should improve when increasing the dimension of the reduced basis.

Let us try to understand this behavior. We recall that the IROM consists in identifying the coordinates of the solution (in the reduced basis) from the sensors temperature. This means that the determined coordinates aim to project the temperature **coming from sensors** on the reduced basis constructed during learning. In order to verify this fact, we plot on Figure 5.12 the maximum error of the sensors temperature reconstruction for various values of retained modes.

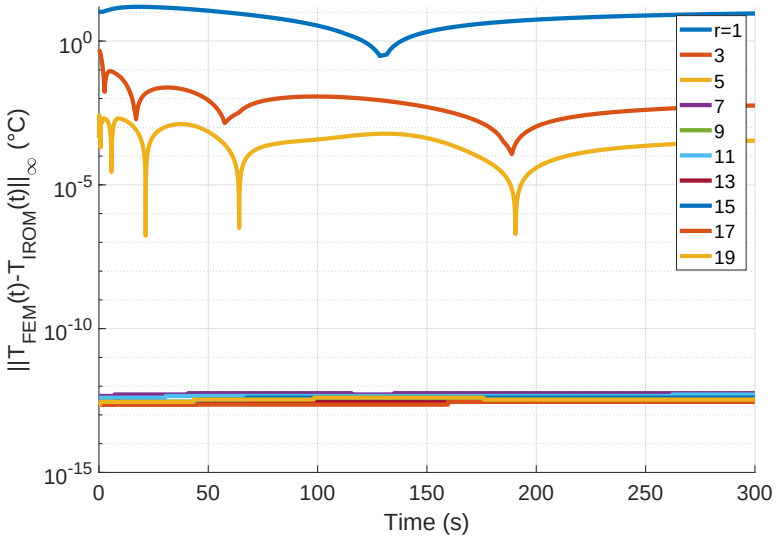
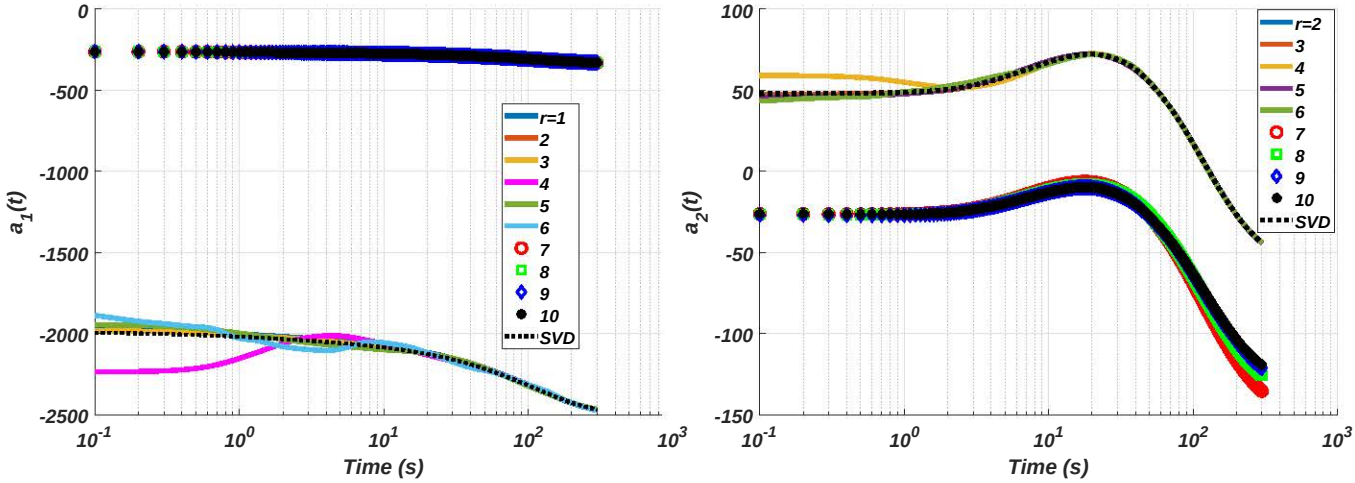


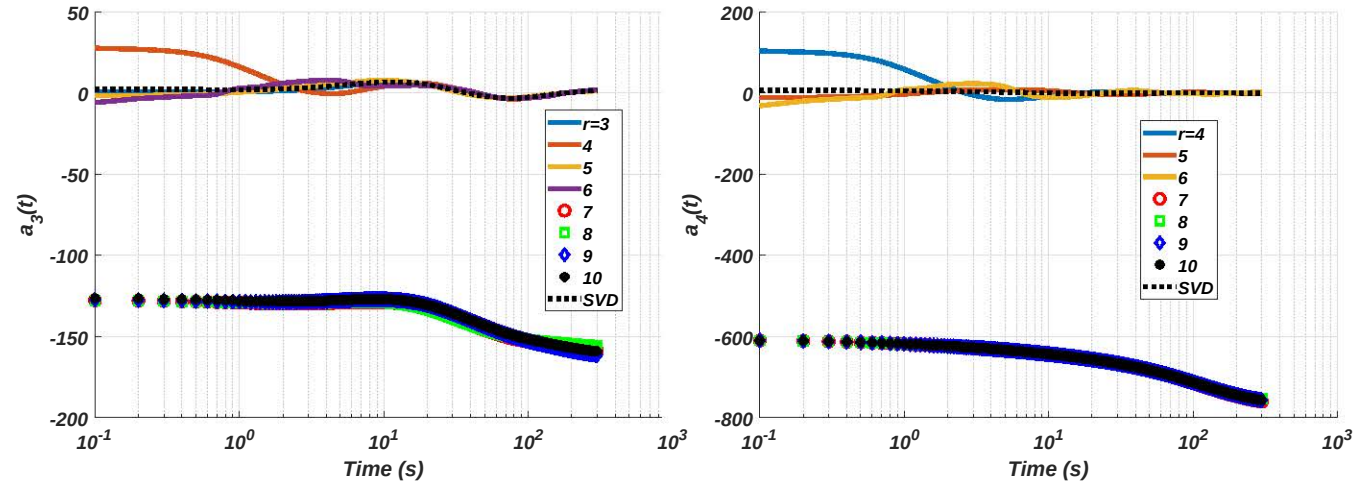
Figure 5.12 – The maximum error of the IROM over the temperature sensors for various number of retained modes $r = 1, \ldots, 19$

It can be seen from Figure 5.12 that increasing the number of retained modes, improves the reconstruction quality of the temperature sensors. This result is expected because the sensors temperature vector belongs to $\mathbb{R}^6$ and when the dimension of the reduced basis increases it becomes more representative of $\mathbb{R}^6$.

If the coordinates of the solution (MOSFETs **and** sensors temperature) are unique, then the identified temporal amplitudes should be identical to the ones obtained from SVD during learning.

In order to investigate such assumption the temporal amplitude of the baseline SVD and those obtained from the IROM are compared in Figure 5.13, for various number of retained modes.





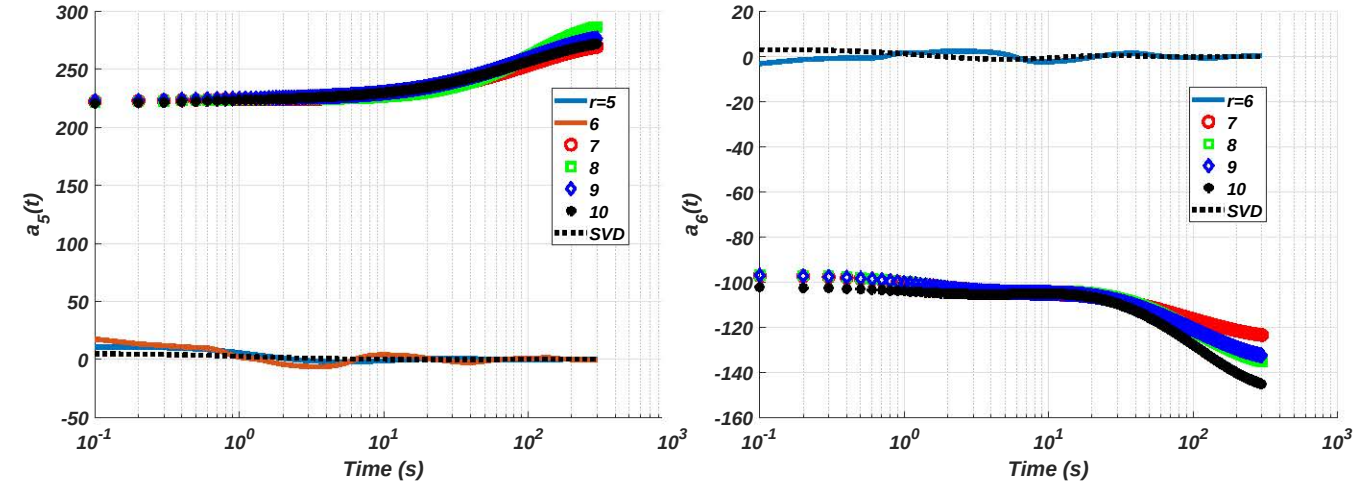


Figure 5.13 – Temporal amplitude obtained by SVD and by IROM for various values of retained modes

It can be seen that increasing the number of retained modes does not improve the IROM accuracy. When the number of retained modes becomes greater than the number of sensors, the identified temporal amplitudes become different from those of the SVD, despite the convergence of the minimization algorithm(difference between the online sensors temperature and its reconstruction in the reduced basis presented in Figure 5.12).

Such behavior can be explained by the fact that an under-determined system has no solution or has an infinity of solutions [12]. Therefore, the number of retained modes should be less or equal to the number of sensors.

It can be seen from Figure 5.10 that the IROM consisting of 3 POD modes has the best accuracy. This fact can be explained by the non-uniqueness of the temporal amplitude obtained by the least square method. The solution is unique if the matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$ is invertible. The optimal number of retained modes regarding this issue can be predicted by computing the determinant of the matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$. The determinant of this matrix for many number of retained modes is presented in the table 5.1.

Number of retained modes r	$det\left((\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)\right)$
1	0.1324
2	0.0187
3	0.0024
4	$1.9\ 10^{-7}$
5	$6\ 10^{-10}$
6	$1.4\ 10^{-15}$
7	$8\ 10^{-28}$
8	$1.6\ 10^{-44}$

Table 5.1 – The effect of the number of retained modes on the determinant of the matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$

It can be noticed that beyond 3 retained modes the determinant becomes less or equal to 10^{-7} which means that the uniqueness of the solution is lost. This fact can be confirmed by the behavior of temporal amplitude presented in Figure 5.13.

Another explanation for this behavior can emerge from the temporal behavior of the sensors temperature presented in Figure 5.14. Figure 5.14 shows that the temporal evolution of the NTCs temperature in the same power module leg are very similar. Therefore, the number of retained modes should be less or equal to the number of independent temperature measurements.

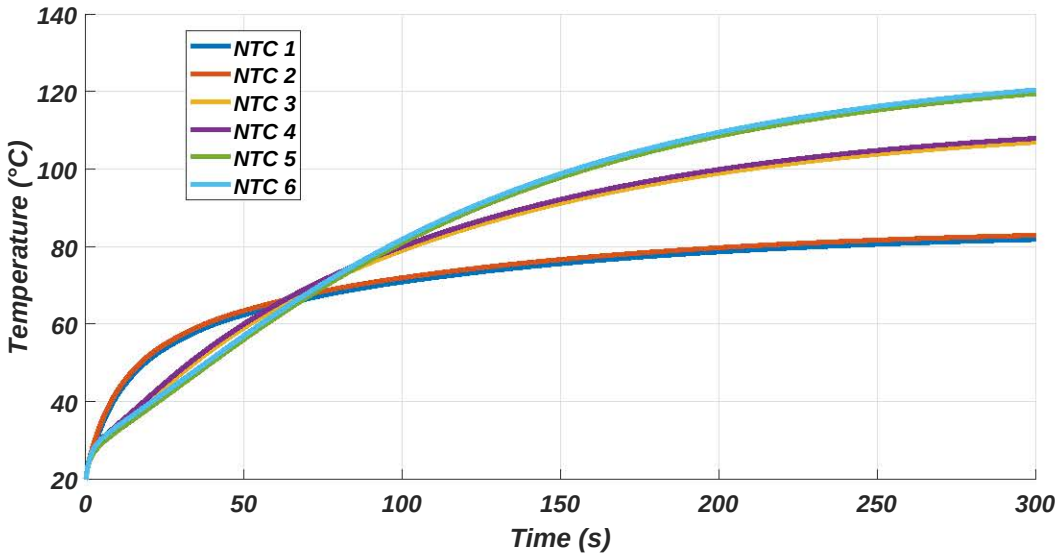


Figure 5.14 – The temporal evolution of the 6 sensors temperature

In conclusion the number of retained modes should respect 3 main requirements:

1. Retain as much as possible the physical information. The information is measured by the magnitude of the singular values.

2. The number of retained modes should be less or equal to the number of sensors.

3. The number of retained modes should ensure a non-zero determinant of the matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$. This condition can be ensured by considering the number of independent temperature measurements in the IROM.

In the following section an IROM of 3 modes will be used.

5.5.2.3 Effect of different operating conditions on the accuracy of the IROM

The question that arises is :

does the IROM remains accurate when usage operating conditions are different from those considered for the construction of the reduced basis ?

To assess the robustness of the IROM regarding this issue, three cases will be studied and in each one an input parameter (fluid inlet velocity, fluid inlet temperature, power losses) is changed with respect to the learning stage.

The parameters used for this task are presented in Figure 5.15.

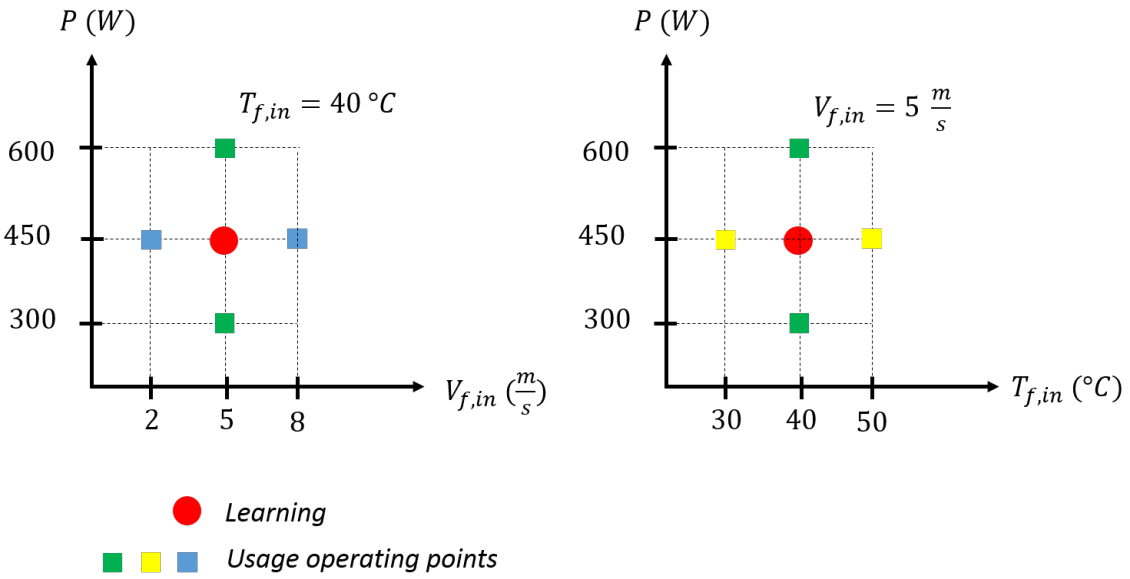


Figure 5.15 – The learning and usage map

5.5.2.3.1 Effect of changing the inlet fluid temperature

In this case, the inlet air temperature is changed while the other parameters are kept the same as in the learning case. Two values of the fluid inlet temperature have been used in this section: $T_{f,in} = 30\ °C$ and $T_{f,in} = 50\ °C$.

The maximum error of the IROM is presented on Figure 5.16.

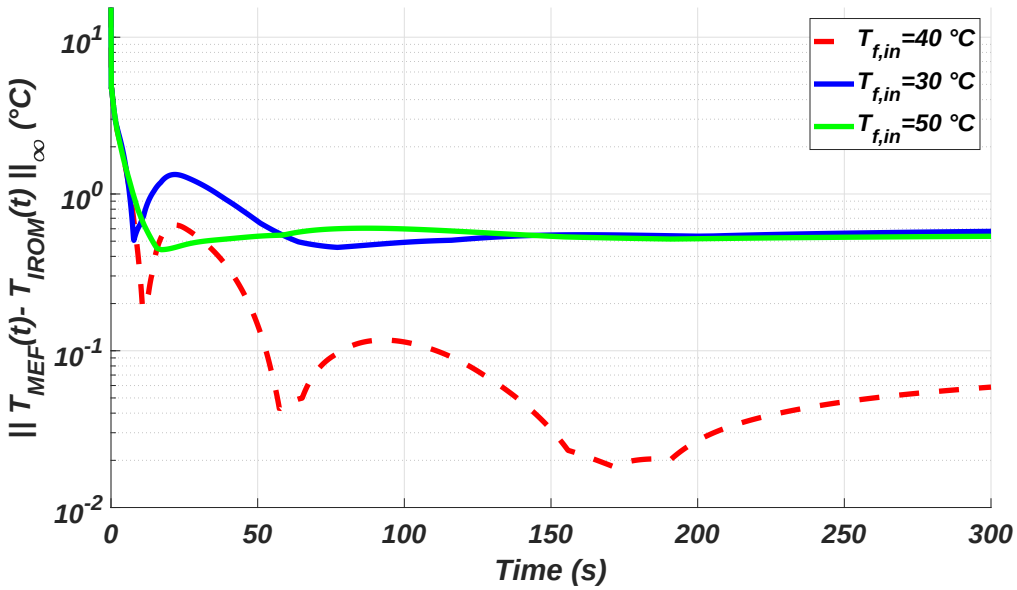


Figure 5.16 – The IROM error when changing the inlet fluid temperature

It can be seen from Figure 5.16 that the error is important at the starting of the transient phase because of the brutal variation of temperature at $t = 0$. Beyond this period, the IROM becomes accurate and the error does not exceed $0.6\ °C$.

5.5.2.3.2 Effect of changing the inlet fluid velocity

In this case, the velocity has been changed with respect to the learning conditions while keeping the same values of the power losses and inlet fluid temperature. The usage velocity values used in this section are $V_{f,in} = 2\ m/s$ and $V_{f,in} = 8\ m/s$. The maximum error is presented in Figure 5.17.

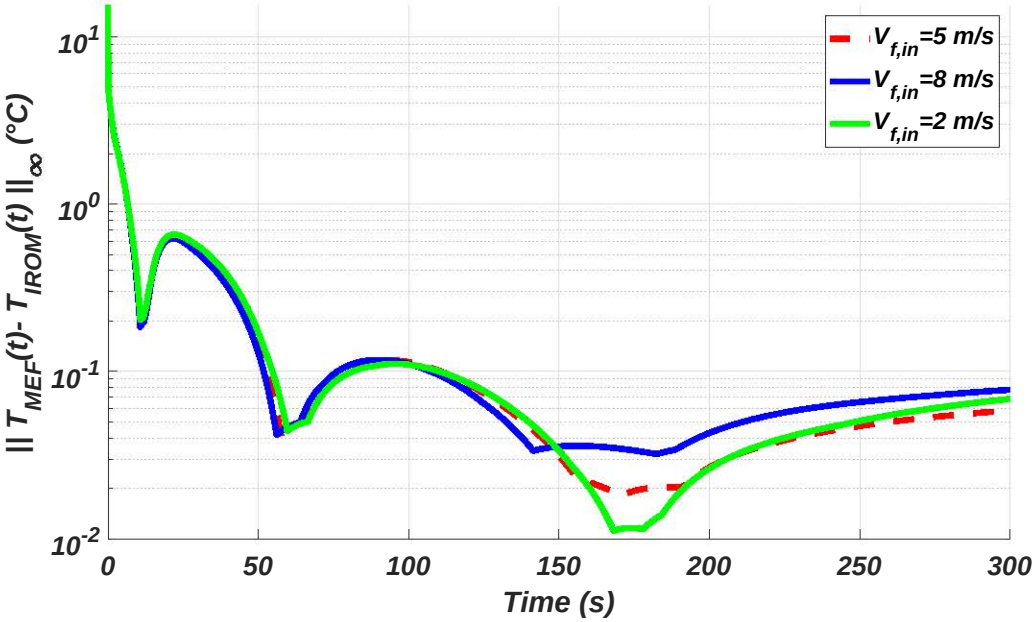


Figure 5.17 – The IROM error when changing the inlet fluid velocity

The same error behavior as in the previous case can be seen when changing the inlet fluid velocity with respect to learning. The IROM realizes a good accuracy beyond the first seconds of the transient period.

It can be seen that the IROM accuracy is less sensitive to the inlet fluid velocity as to the inlet fluid temperature.

5.5.2.3.3 Effect of changing the power losses

In this case, the power losses have been changed with respect to learning while keeping the same values of the inlet fluid temperature and velocity. The usage power losses values used in this section are $P = 300\ W$ and $P = 600\ W$. The error is presented in Figure 5.17.

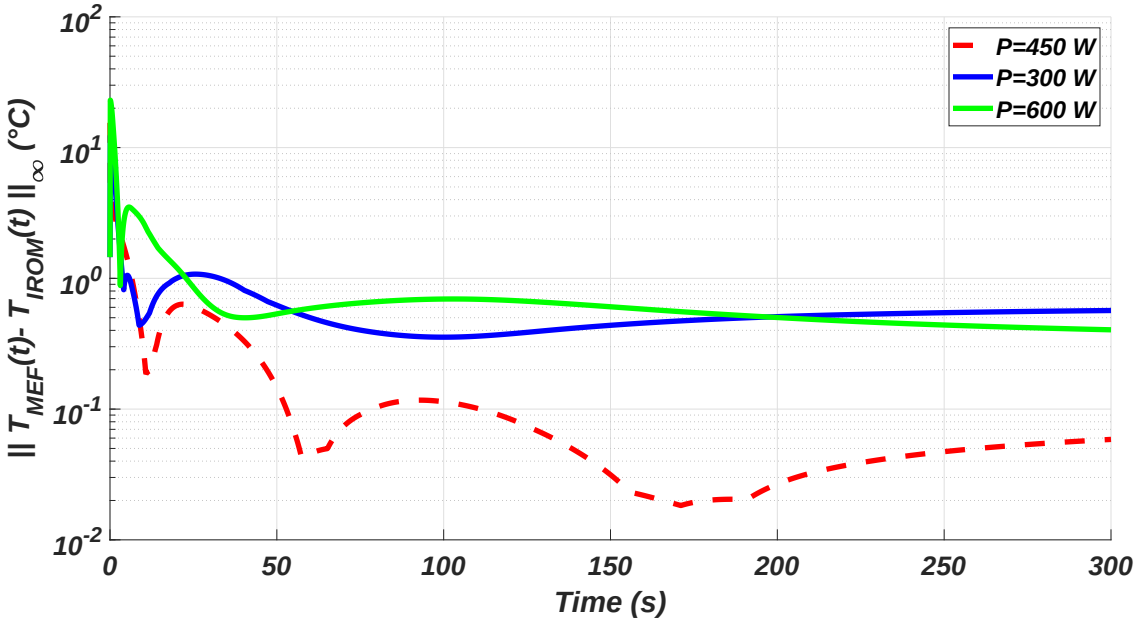


Figure 5.18 – The IROM error when changing the power losses

It can be seen from Figure 5.18 that changing power losses with respect to learning produces an important error during the first $5\ s$. Beyond $50\ s$ this error becomes lower than $0.5\ °C$.

5.5.3 Effect of sensors placement and number on the accuracy of the IROM

In section 5.5.2.2, the effect of the number of retained modes has been adressed in the case of a fixed number of temperature sensors. The performance of the IROM when other measurement locations can be added is not studied, which is the aim of this section. The additional measurement locations for a single phase of the power module are presented in Figure 5.19. In this study, the usage operating conditions are identical to the learning operating conditions. The considered operating conditions are $P = 450\,W$, $T_{f,in} = 40\ °C$ and $V_{f,in} = 5\,m/s$.

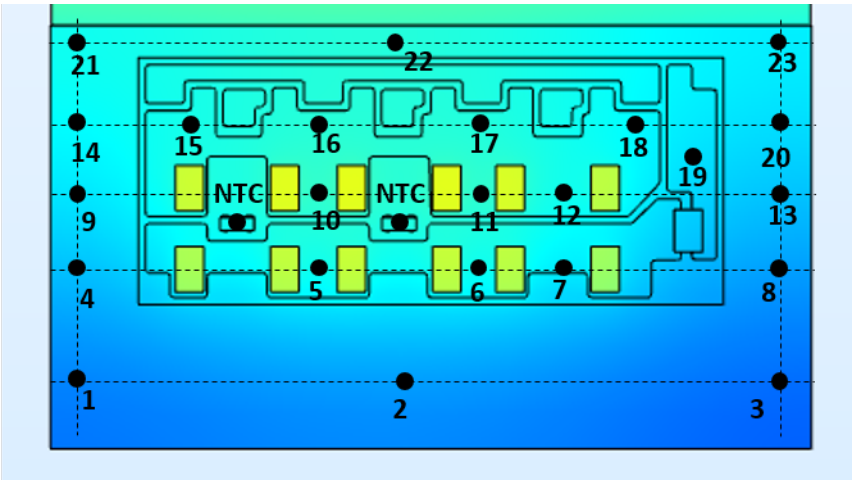


Figure 5.19 – Measurement locations for one leg

The temperature of each of these additional 23 locations is recovered in the snapshots matrix. The mean temperature of NTC sensors and the maximum temperature for each MOSFET are recovered. Therefore, $(23$ measurement locations$+2$ NTCs$+12$ MOSFETs$) \times 3$ legs$= 111$ temperatures are recovered for the three legs power module in order to construct the snapshots matrix. The SVD is performed on the snapshots matrix and the singular values and relative energy content distributions are presented in Figure 5.20 and 5.21.

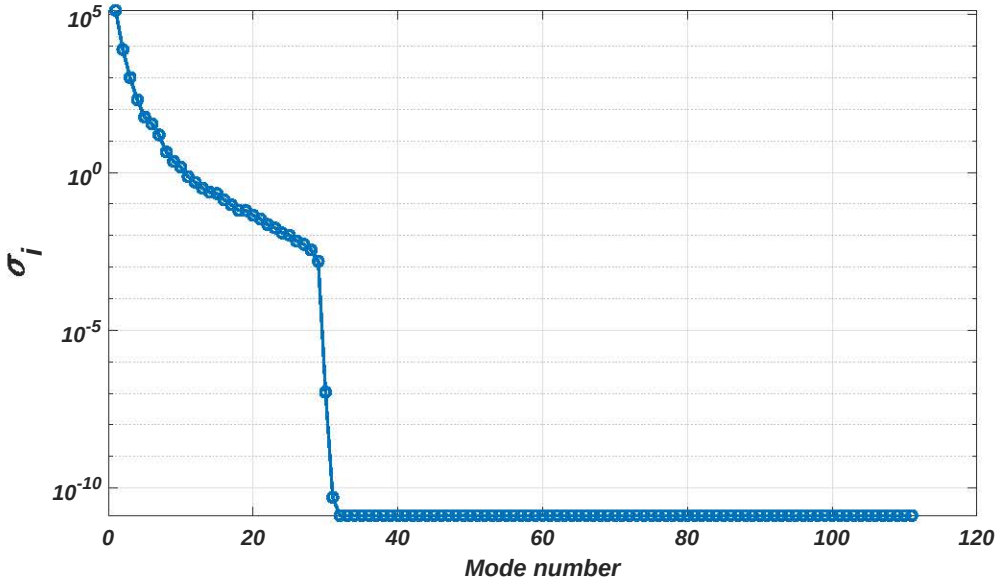


Figure 5.20 – Singular values distribution

The relative energy content is presented in Figure 5.21.

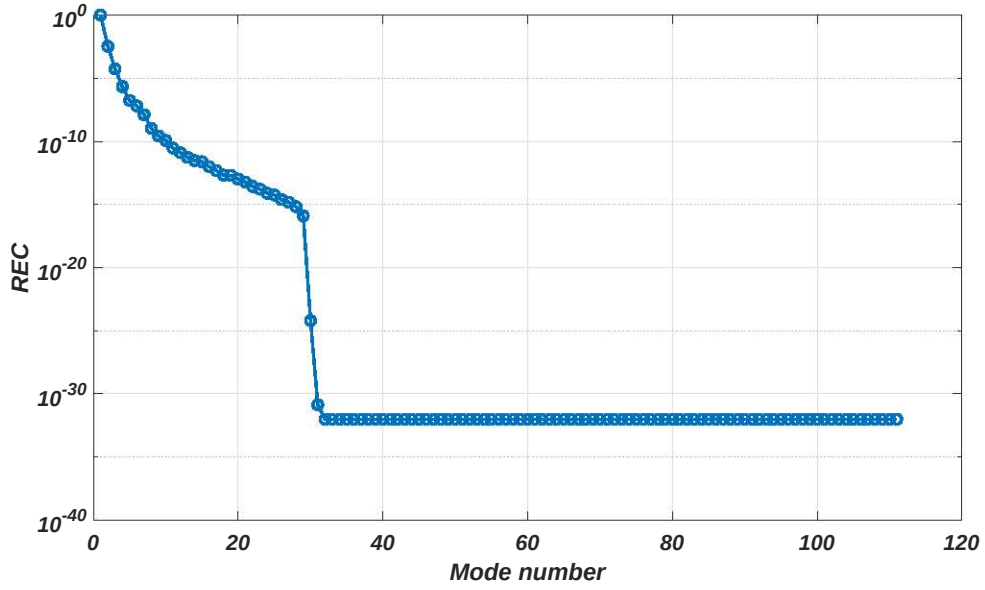


Figure 5.21 – Relative Energy Content (REC) distribution

There are several ways to place the temperature sensors on the power module. If we want to keep k measurement locations among n allowed locations, there exists $C_n^k = \frac{n!}{k!\,(n-k)!}$ possible ways to place the temperature sensors. For instance, there exist $C_{24}^8 = 735471$ ways to place 8 sensors among 24 allowed locations, which is a huge number of possibilities that will require important computational resources.

Therefore, an heuristic approach to select the locations of the sensors consists in keeping the independent measurements. In our case, 6 arrangements have been considered and are presented in Figure 5.22 (red dots). It should be outlined that the choice of sensors locations is not unique and other optimal locations can be found.

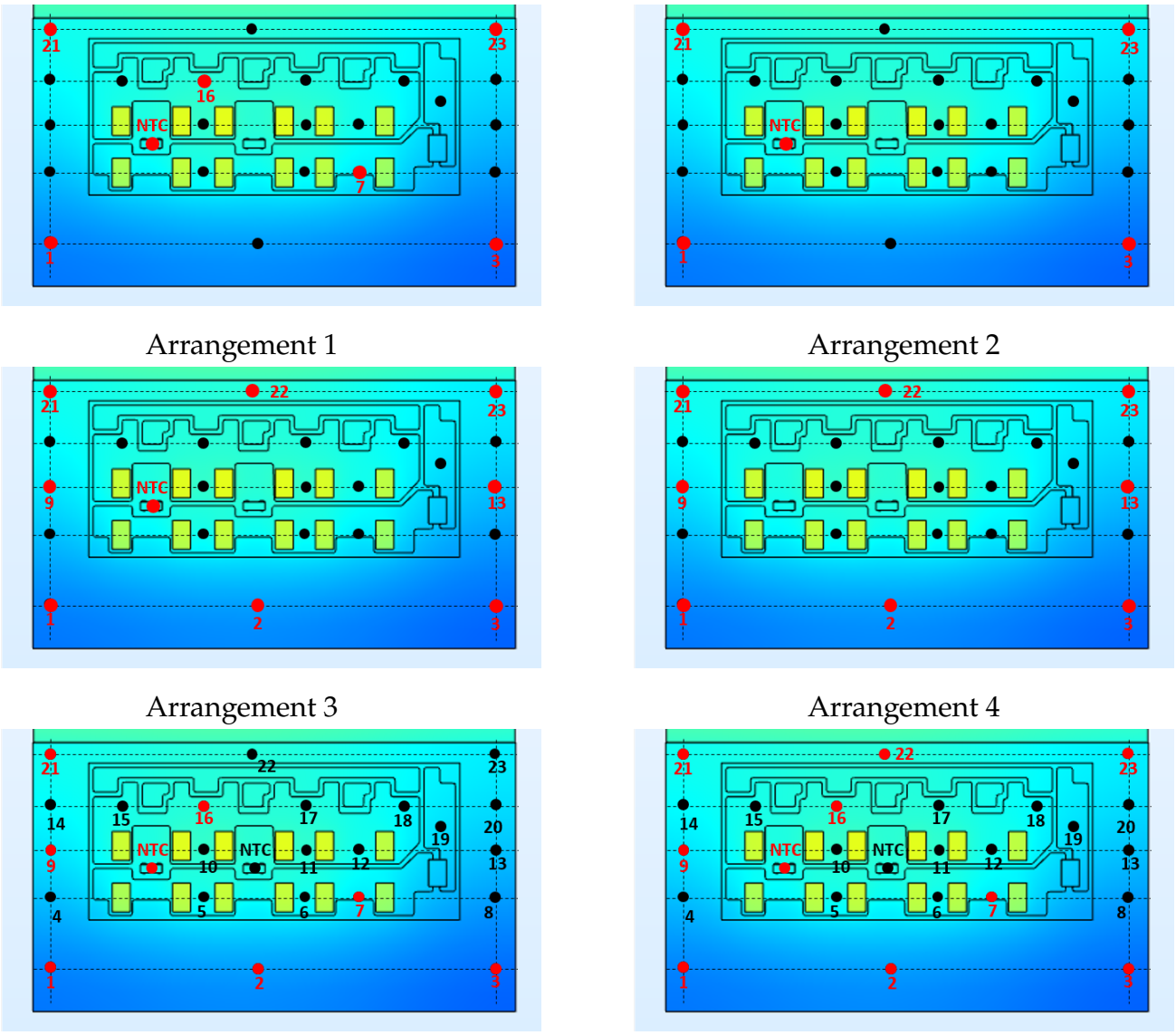


Figure 5.22 – Different temperature sensors placements

The temporal evolution of the temperature in the measurement locations for each of the sensors arrangement is presented in Figure 5.23.

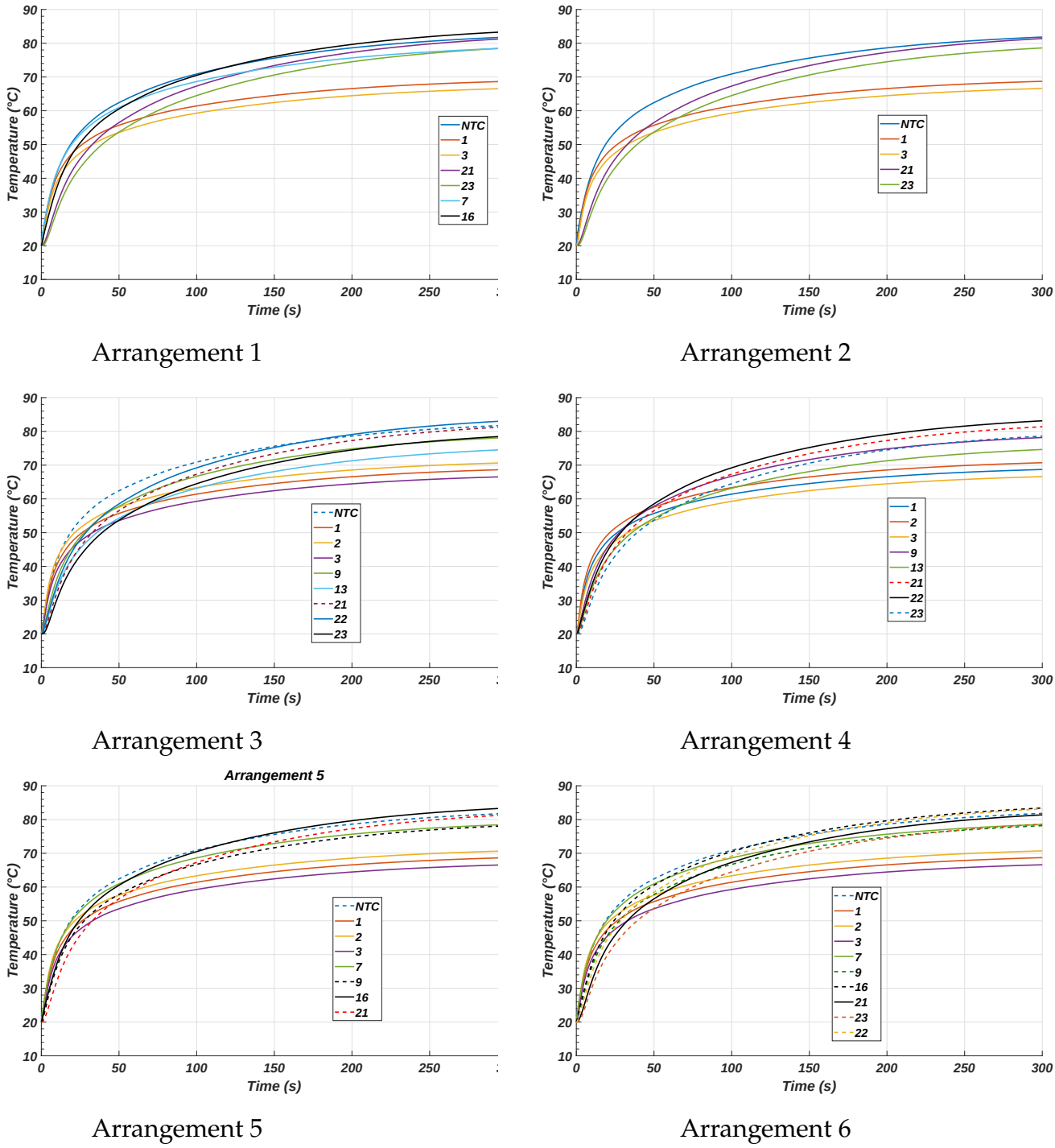


Figure 5.23 – The temperature evolution in the temperature sensors for different arrangement

It can be seen from Figure 5.23 that for each arrangement the temperature evolution of the sensors are independent, as the tangent line sweeps a large range of characteristic times $\left(\frac{1}{T(t)}\frac{dT(t)}{dt}\right)^{-1}$.

For each of the previous arrangements, the number of retained modes has been chosen equal to the number of temperature sensors used for the three legs power module. For instance, in the first arrangement $r = 7 \times 3(\text{legs}) = 21$ POD modes have been retained.

The reconstruction error using the first r POD modes is bounded by $\sqrt{min(M,N)-r}\,\sigma_{r+1}$ (see equation (3.37)). The reconstruction error bound for each of the sensors arrangement is depicted in Table 5.2.

Arrangement	Number of measurement locations per phase	number of retained modes r	Reconstruction error bound ($\sqrt{min(M,N)-r}\,\sigma_{r+1}$)
1	7	21	$\sqrt{111-21}\,\sigma_{22} = 0.2\,°C$
2	5	15	$1.4\,°C$
3	9	27	$0.03\,°C$
4	8	24	$0.08\,°C$
5	8	24	$0.08\,°C$
6	10	30	$2\,10^{-10}\,°C$

Table 5.2 – The reconstruction error bound for different sensors arrangements

It can be seen from Table 5.2 that increasing the number of retained modes, decreases the reconstruction error. For the arrangement 6, the reconstruction error have the same order of magnitude as the round-off error.

The matrix **A** (equation (3.70)) relating the MOSFETs temperature to the measurement temperatures have been constructed for each sensors arrangement.

The accuracy of the IROM has been assessed for each sensors arrangement and the maximum norm of the difference between the IROM and the FEM have been plotted in Figure 5.24.

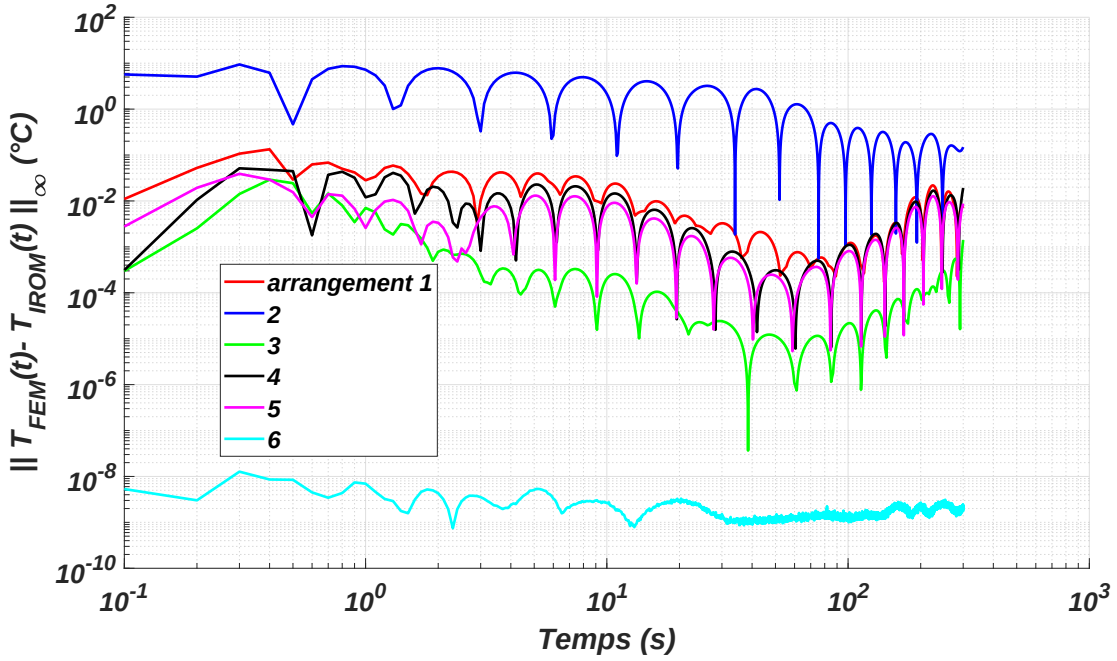


Figure 5.24 – The IROM error for different temperature sensors arrangements

It can seen from Figure 5.24 that arrangement 2 has the worst accuracy. Therefore adding temperature sensors at the corners of the heat spreader is not enough in order to improve the IROM accuracy.

In arrangement 1, additional measurement locations have been added on the circuit board (measure locations 16 and 17), which have improved significantly the accuracy of the IROM. In this case the IROM error becomes bounded by $0.2\,°C$.

In arrangement 3, temperaure measurement locations have been added at the middle edges of the heat spreader with respect to the arrangement 2. Such choice, improves the IROM accuracy and the IROM error becomes lower than $0.04\,°C$, which is very satisfactory.

In arrangement 4, all internal measurement locations (on the circuit board) have been removed with respect to arrangement 3. This case reveals interesting performance and the maximum error is lower than $0.05\,°C$. This result implies that there is no need to place temperature sensors on the electronic board in order to track the temperature of MOSFETs.

In arrangement 5, temperature sensors are placed on the two edges of the heat spreader and three internal measurement locations (NTC, locations 16 and 7) are considered. The accuracy of such arrangement is similar to arrangement 4 which can be explained by the fact that the number of retained modes is the same between the two cases.

In arrangement 6, 10 measurement locations have been considered as depicted in Figure 5.22. This configuration realizes the best accuracy and the error is lower than $10^{-8}\,°C$. In this arrangement $r = 30$ POD modes are retained which correspond to the abrupt decrease of singular values as can be seen from Figure 5.20.

It can be seen from Figure 5.24 that arrangements 1 and 4 have roughly the same accuracy. In arrangement 1, 4 external measurements locations (outside the electronic board) and 3 internal measurements locations have been considered. In arrangement 4, only 8 external measurements have been chosen. Therefore, 4 external measurements can realize the same accuracy as 3 internal measurements. From a practical point of view, it is easier to place temperature sensors on the heat spreader than placing them on the circuit board.

In the previous case, the POD basis have been computed using all the allowed measurement locations. The question that arises is: does the IROM accuracy remains the same if we use only the selected measurement locations (to construct the snapshots matrix) in order to construct the POD basis? For this reason, we consider the arrangement 2, and we compare the two approaches:

1. The POD basis is computed only once using all allowed measurement locations, as in the previous case. This procedure allows to test many selected measurement locations using the same POD basis.

2. The POD basis is computed from the selected measurement locations and the information that we aim to determine (MOSFETs temperature). The drawback of such procedure is that a SVD has to be performed for each chosen measurement locations, which can be very time consuming if we want to test a large number of sensors arrangements.

The maximum norm of the difference between the IROM and the FEM has been plotted for the previous two cases in Figure 5.25.

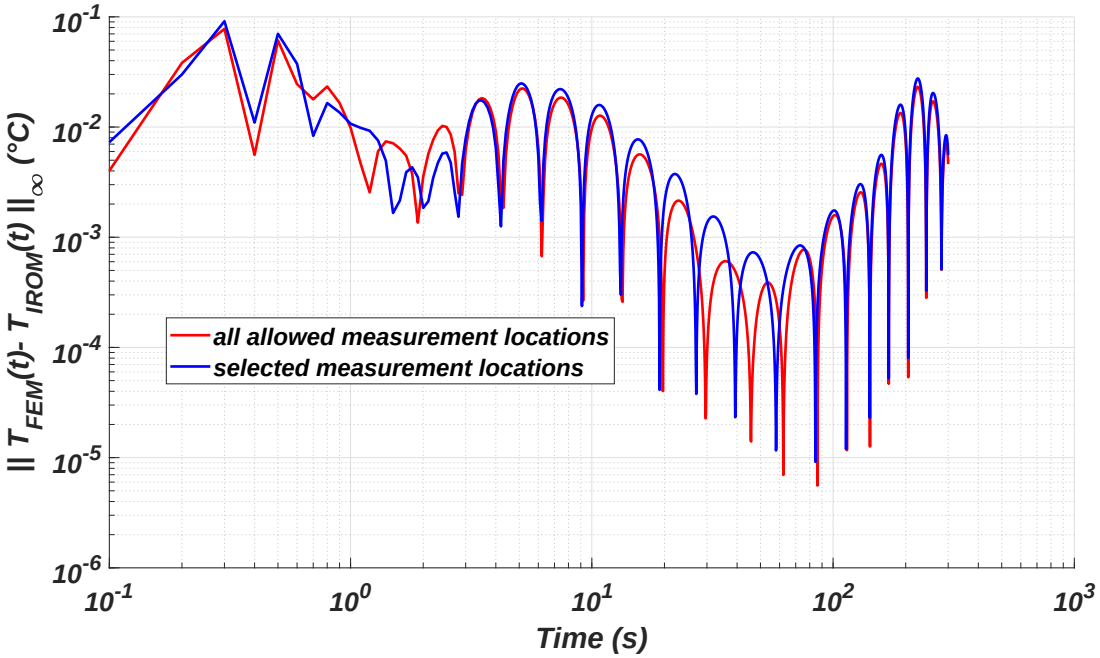


Figure 5.25 – The IROM error when constructing the POD basis using all measurement locations and selected measurement locations

It can be seen from Figure 5.25 that the two models have roughly the same accuracy. From a practical point of view, we do not need to construct a POD basis for each sensor location arrangement. Therefore, a generic POD basis can be computed for all allowed measurement locations and then test many selected ones.

This behavior can be explained from the singular value distribution for the two cases as presented in Figure 5.26.

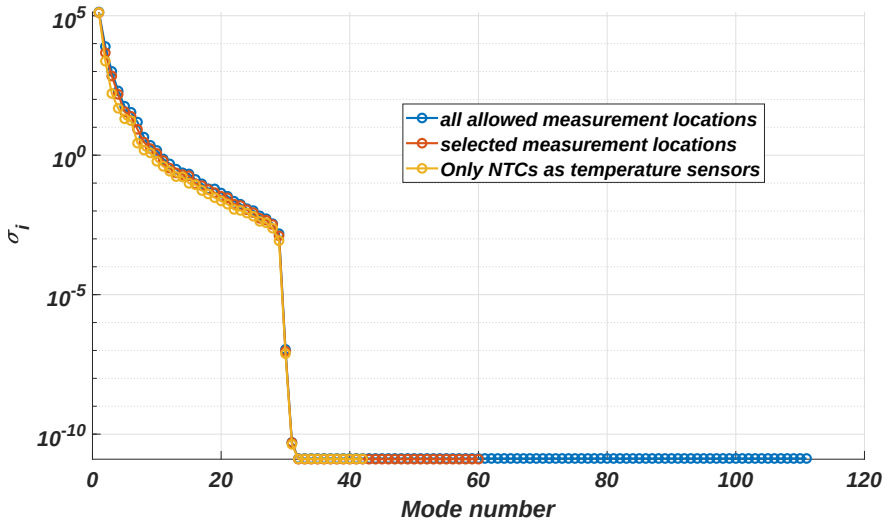


Figure 5.26 – The singular values distribution for various measurement arrangements

It can be seen from Figure 5.26 that the singular values destribution is roughly the same for

the two cases with a slight difference in the magnitude of the singular values. The difference in magnitude is due to the fact that the snapshots matrix with all measurement locations contains a larger amount of data.

The singular values distribution when only NTC sensors are considered is also plotted in Figure 5.26. The singular values distribution has the same behavior as the previous two cases. In section 5.5.2.1, only $r = 5$ POD modes have been retained because the number of sensors should be larger than the number of retained modes. This can explain the lower accuracy of the IROM presented in section 5.5.2.1.

5.6 Conclusions

In this chapter, an application of the inverse reduced order model has been given for the three-legs power electronic module. The two stages of the IROM, namely the learning and the usage stages, have been depicted. It have been shown that the number of dominant modes should be lower than the number of sensors and should ensure a non-zero determinant of the matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$ in order to identify the right temporal amplitude. The IROM constructed in this chapter consists of 3 dominant modes which is convenient for real time temperature monitoring in the electronic power module.

The only input information required for the IROM is the temperature sensors placed far from MOSFETs. The IROM does not need to know the boundary conditions or power losses in the power model, which represents the novelty of such approach. The accuracy of the IROM when changing the operating conditions with respect to the learning stage has been studied. Typically, obtained results show that the IROM remains accurate even when usage operating conditions are different from the learning ones. The IROM accuracy has shown to be sensitive to power losses and inlet fluid temperature variation with respect to learning, while it remains unaffected when submitted to a different inlet fluid velocity.

Measurement locations have been added to the IROM and obtained results show an improvement of the IROM accuracy, in particular during sharp variations of power losses. Several measurement arrangements have been chosen heuristically and the accuracy of each one has been studied. For instance, when using 8 external temperature sensors per leg, the maximum absolute error is less than $0.05\,^{\circ}C$.

Bibliography

[1] Ho-Ming Tong, Yi-Shao Lai, and CP Wong. *Advanced flip chip packaging*. Springer, 2013.

[2] FN Masana. A new approach to the dynamic thermal modelling of semiconductor packages. *Microelectronics Reliability*, 41(6):901–912, 2001.

[3] Vladimír Székely and Tran Van Bien. Fine structure of heat flow path in semiconductor devices: a measurement and identification method. *Solid-State Electronics*, 31(9):1363–1368, 1988.

[4] Mauro Ciappa, Wolfgang Fichtner, T Kojima, Y Yamada, and Y Nishibe. Extraction of accurate thermal compact models for fast electro-thermal simulation of igbt modules in hybrid electric vehicles. *Microelectronics Reliability*, 45(9-11):1694–1699, 2005.

[5] Guojun Yu, Bo Yu, Dongxu Han, and Liping Wang. Unsteady-state thermal calculation of buried oil pipeline using a proper orthogonal decomposition reduced-order model. *Applied Thermal Engineering*, 51(1-2):177–189, 2013.

[6] Adam Fic, Ryszard A Białecki, and Alain J Kassab. Solving transient nonlinear heat conduction problems by proper orthogonal decomposition and the finite-element method. *Numerical Heat Transfer, Part B: Fundamentals*, 48(2):103–124, 2005.

[7] Hazem Ben Aissia, Jacques Jay, Ronnie Knikker, and Shihe Xin. Modèle thermique réduit appliqué à une carte électronique de puissance. In *Société Française de Thermique*, 2017.

[8] SR Carvalho, SMM Lima e Silva, AR Machado, and G Guimaraes. Temperature determination at the chip–tool interface using an inverse thermal model considering the tool and tool holder. *Journal of Materials Processing Technology*, 179(1-3):97–104, 2006.

[9] Manuel Girault and Daniel Petit. Identification methods in nonlinear heat conduction. part i: Model reduction. *International Journal of Heat and Mass Transfer*, 48(1):105–118, 2005.

[10] Manuel Girault and Daniel Petit. Identification methods in nonlinear heat conduction. part ii: inverse problem using a reduced model. *International Journal of Heat and Mass Transfer*, 48(1):119–133, 2005.

[11] Etienne Videcoq, Myriam Lazard, Olivier Quemener, and Alain Neveu. Online temperature prediction using a branch eigenmode reduced model applied to cutting process. *Numerical Heat Transfer, Part A: Applications*, 55(7):683–705, 2009.

[12] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Summary and Outlook

Conclusions

This thesis focuses on the construction of two reduced order models for the purpose of monitoring the temperature in a power electronic module. The first reduced order model is called direct because it has as input parameters the boundary conditions and source terms (such as power losses, air inlet temperature and velocity) allowing the direct resolution of the thermal problem. The second reduced order model is called inverse because it has as input parameters only the sensors temperature located far from the electronic components (MOSFET transistors). These two models will deal with any practical situation regarding the available information.

Each reduced order model consists of two main stages :

1. **The learning stage :**

   In this stage, a reduced basis that represents accurately the temperature field is constructed. The first step of this stage is to run the finite element model for given operating condition (boundary conditions, thermal load, ...). This step can be time-consuming because the finite element model has a dimension equal to the number of mesh nodes. The obtained results are stored in a snapshots matrix. Each column of this matrix contains the three-dimensional temperature field at a given instant t_i. The singular value decomposition (SVD) is then applied on the snapshots matrix. This decomposition allows to represent each snapshot (the temperature field at t_i) as a linear combination of the proper orthogonal decomposition (POD) modes. The number of POD modes is equal to the number of non-zero singular values of the SVD. The contribution of each POD basis component is not the same. Therefore, a selection procedure should be carried out in order to keep the r dominant modes. This stage is done offline and only once because the SVD is time-consuming. The dominant modes (reduced basis) are the output of this stage that will be used to construct the reduced order model.

2. **The usage stage :**

   In this stage, the coordinates of a new solution will be computed in the reduced basis that has already been constructed during the learning stage. The usage operating conditions in this stage can be different from those used in the learning stage. This stage is performed in real time.

   For the direct reduced order model (DROM), the coordinates of the new solution are obtained from the Galerkin projection of the finite element model on the reduced basis. For the inverse reduced order model (IROM), the coordinates of the new solution are obtained from the sensors temperature, because they have the same coordinates as the MOSFETs temperature.

The DROM has been constructed for both the steady and the transient states.

The steady state DROM of order 4 has been shown accurate for large range of power losses and convection coefficient. This model realizes a computing time gain of 267 and an error that does not exceed $0.03\ °C$. The effect of the number of retained modes on the accuracy of this model has been investigated and confirms the relevance of the dominant modes selection criteria.

The transient DROM has been constructed using the first 10 POD modes. This model has shown to be accurate for a power profile containing sharp time variations and a large range of amplitudes. The error generated by this model is less than $0.1\ °C$ (except during the first second of sharp temporal variation of the dissipated power). Such model allows to decrease drastically the computing time from 11 hours to 1.7 seconds for the studied case. The effect of the number of retained modes on the accuracy has been studied. Obtained results reveal that high-order POD modes are responsible for describing the sharp time variation of the temperature.

The accuracy of the latter model has been investigated with respect to the snapshots time step. It has been found that increasing the snapshots time step enhances the accuracy of the DROM during the steady state and decreases it during the transient phase.

The IROM has been constructed for the three legs power module. The novelty of this model is its capability to compute the electronic components temperature by just knowing the sensors temperature. It has been shown that the number of retained modes should verify two requirements: first, to be less or equal to the number of sensors and secondly to realize a non singular matrix $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$ (presented in Chapter 3). These requirements allow to identify the physical solution coordinates in the reduced basis. Obtained results showed that the accuracy of the IROM beyond the first seconds is satisfactory. The accuracy of the IROM has been investigated when changing the usage operating conditions (the power losses, the inlet air temperature and velocity) with respect to learning. Obtained results show that the accuracy is more sensitive to a change in the power losses or the cooling fluid inlet temperature than a change in the cooling fluid inlet velocity, with respect to learning operating conditions. Additional measurement locations have been added to the IROM and obtained results show an improvement of the IROM accuracy, in particular during the first seconds. Several measurement arrangements have been chosen heuristically and the accuracy of each one has been tested. For instance, when using 8 external temperature sensors per leg, the maximum absolute error is less than $0.05\ °C$.

Outlook

Many enhancements can be suggested for this work:

1. Study the effect of changing the convection coefficient and the cooling fluid temperature with respect to learning operating conditions on the accuracy of the transient DROM. In fact, the reduced basis is computed for given operating conditions and its capability to represent other ones is not guaranteed. In our study, only the power losses have been changed between the learning and the usage stage.

2. Improve the modeling of the heat transfer coefficient and the fluid temperature for the DROM by considering their space variations. In fact, when the air crosses the heat fins,

it heats up and its velocity profile changes in the flow direction which results in a non-uniform heat transfer coefficient and fluid temperature.

Such improvement can be achieved by performing learning from conjugate heat transfer numerical simulations or experimental measurements. In this stage, a POD basis can be derived in order to establish a coupling between the unknown information (space variation of the convection coefficient and the fluid temperature) and the known information (mass flow rate, inlet fluid temperature, sensors temperature). For new operating conditions, the unknown information can be obtained using the mathematical formulation of the IROM.

3. Treat the case of a power losses that is not uniformly distributed between MOSFETs. In fact, the on-state electrical resistance of MOSFETs are not identical which results in a non-uniform power losses in the power module.

   To deal with such situation, a POD basis can be computed for each case of a single turned-on MOSFET. Then the DROM will be computed for each of these cases. The resulted temperature field can be computed by applying the superposition principale.

4. Mount an experimental bench in order to validate the numerical model. In fact, many assumptions are made in the numerical model regarding the values of the contact thermal resistance, the boundary conditions, etc. Calibration methods can be used to make the numerical results in good agreement with experimental measurements.

5. Broaden the study of the effect of snapshots time steps on the accuracy of the DROM in order to find the optimal learning time step for a chosen usage time step. This can guide us to choose the optimal snapshots time step if the user wants to run the DROM with a different time step.

6. Determine the optimal sensor locations for the IROM. A possible way to reach this target is to consider the set of all allowed sensor locations. Instead of testing all combinations, an error estimator can be constructed for the inverse reduced order model and can be expressed as a function of the sensors selection matrix ($\mathcal{S}_{kn}$). We have started working on this estimator but numerical validation has not been performed yet.

7. Construct a priori error estimator as a function of the usage and the learning operating conditions. It has been shown in the DROM that the accuracy is affected by the change in operating conditions with respect to learning. The resulted error is determined after launching the finite element simulation and computing its deviation with respect to the reduced order model. This error is computed after running the model that is why it can be described as a posteriori.

8. Interpolate the POD modes in space in order to simulate differents dimensions of the considered geometry. In fact, the POD modes are defined for a given mesh that depends on the geometry. When changing the dimension of the geometry (for instance the fin's thickness) the old POD basis cannot be used for the new geometry (mesh). Classical interpolation (Lagrange interpolation) cannot be used because it does not preserve the orthogonality of the basis. Advanced interpolation methods can be worked out to meet this need.

Appendix A

Straight heat fins calculation

When conducting a thermal study on heat sink, modeling the energy exchange between the cooling fluid and fins involves two coupled physics: fluid mechanics and heat transfer. In fact, determining the temperature field in the heat sink requires solving the Navier-Stokes equations and the energy equation in the fluid (cooling fluid) part and coupling it to the heat equation in the solid part (fins). Such coupled heat transfer problem is called conjugate heat transfer (CHT). The aim of this part is to model the heat sink by an equivalent heat transfer coefficient. A demonstration of how to compute such coefficient will be presented as well as the physical assumptions that allow to obtain it.

A.0.1 Geometry description

The heat sink geometry description is given in figure A.1.

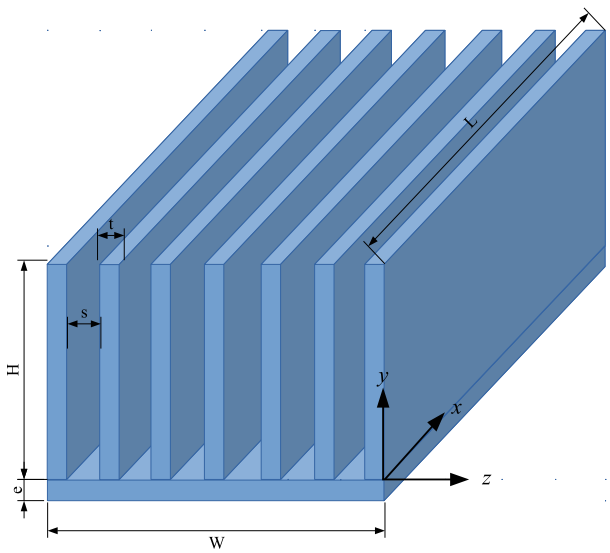


Figure A.1 – Heat sink

Where L, H and t represent respectively the length, the height and the thickness of a single heat fin. s denotes the spacing between two fins. W is the width of the heat sink.

If we denote by N the number of heat fins then we can write:

$$W = (s + t) \times (N - 1) + t$$

A.0.2 Single channel thermal resistance

To calculate the thermal resistance of the heat sink, a single channel is considered. A heat sink can be viewed as an assembly of many channels arranged in parallel with respect to the main heat flux coming from the base plate and hence the heat sink thermal resistance can be deducted. To compute the thermal resistance of a single channel, an energy balance is performed on a control volume located between x and $x + dx$ as shown in figure A.2.

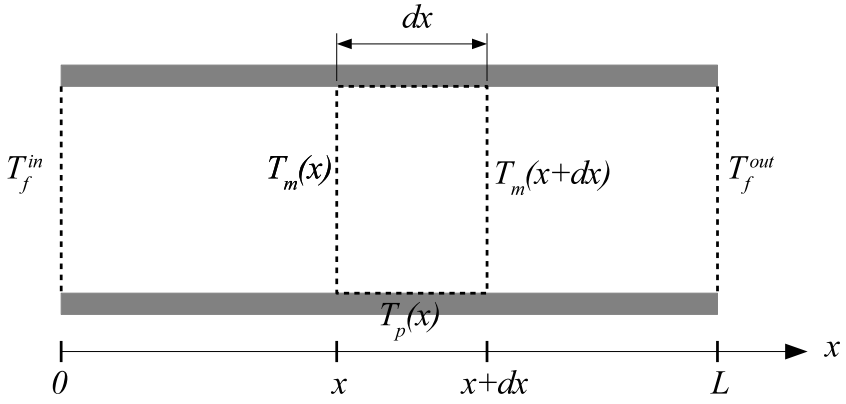


Figure A.2 – Thermal energy balance on a differential element in the channel

Let us define $T_m(x)$ as the mean (or bulk) fluid temperature [1] at position x which is usefull when performing an energy balance. $T_m(x)$ is defined as :

$$T_m(x) = \frac{1}{\dot{m} C_p} \int_A \rho u T C_p dA$$

Where $\dot{m}$ is the mass flow rate, C_p is the fluid thermal capacity, T is the local temperature and u is the x-direction velocity. T_f^{in} and T_f^{out} are the fluid inlet and outlet temperature. A is the area crossed by the fluid flow.
In steady state, the energy balance of the control volume located between x and $x + dx$ can be written as:

$$\Phi_{in} = \Phi_{out} + \Phi_{convection} \tag{A.1}$$

Where

$$\Phi_{in} = \dot{m} C_p T_m(x)$$
$$\Phi_{out} = \dot{m} C_p T_m(x + dx)$$
$$\Phi_{convection} = h(x)(P.dx)(T_m(x) - T_p(x))$$

In the last equation $h(x)$ is the local convection coefficient and P is the wetted perimeter ($P = 2(s + H)$).
By replacing each flux by its expression in equation A.1, the energy balance becomes:

$$\dot{m}C_p\frac{dT_m}{dx} = -Ph(x)(T_m(x) - T_p(x)) \tag{A.2}$$

Uniform wall temperaure

When the wall temperature does not depend on position x, equation A.2 can be integrated analytically. Let us denote by T_p the uniform wall temperature. Dividing equation A.2 by $\dot{m}C_p$ and integrating it between 0 and L yields:

$$\int_0^L \frac{dT_m}{T_m(x) - T_p} = \int_0^L \frac{-Ph(x)}{\dot{m}C_p}dx$$

which gives

$$ln(\frac{T_m(L) - T_p}{T_m(0) - T_p}) = \frac{-PL}{\dot{m}C_p}\left(\frac{1}{L}\int_0^L h(x)dx\right) \tag{A.3}$$

Let us define the mean convection coefficient by $h_m = \frac{1}{L}\int_0^L h(x)dx$ then equation A.3 becomes:

$$\frac{T_f^{out} - T_p}{T_f^{in} - T_p} = exp(\frac{-PLh_m}{\dot{m}C_p}) \tag{A.4}$$

In order to determine the temperature at any position x, equation A.2 must be integrated between 0 and x. Let's define $h_m(x)$ as $h_m(x) = \frac{1}{x}\int_0^x h(\alpha)d\alpha$ then the fluid mean temperature at any position x can be obtained from equation :

$$\frac{T_m(x) - T_p}{T_f^{in} - T_p} = exp(\frac{-Ph_m(x)}{\dot{m}C_p}x) \tag{A.5}$$

We should keep in mind that the last equation is only valid in the case of uniform wall temperature.

We can calculate the efficiency of the fin to assess the temperature gradient with respect to the fin height direction. The efficiency does not give any information about the temperature gradient in the fin along the fluid flow direction.

The heat removed by forced convection is equal to :

$$\Phi_{total} = \dot{m}C_p\left(T_f^{out} - T_f^{in}\right) = \dot{m}C_p\left(T_f^{out} - T_p + T_p - T_f^{in}\right) = \frac{T_p - T_f^{in}}{R_{channel}} \tag{A.6}$$

By inserting equation A.4 in A.6, the thermal resistance between the channel (assumed to be isothermal) and the inlet air temperature is obtained:

$$R_{th}^{channel} = \frac{1}{\dot{m}C_p\left[1 - exp(\frac{-PLh_m}{\dot{m}C_p})\right]} \tag{A.7}$$

A.0.3 Thermal conduction in fins and fin's efficiency

In previous sections, a description of the heat exchange at the interface between the fluid and the solid is given without investigating what happens inside fins. In this part, the temperature evolution within a single fin will be described.

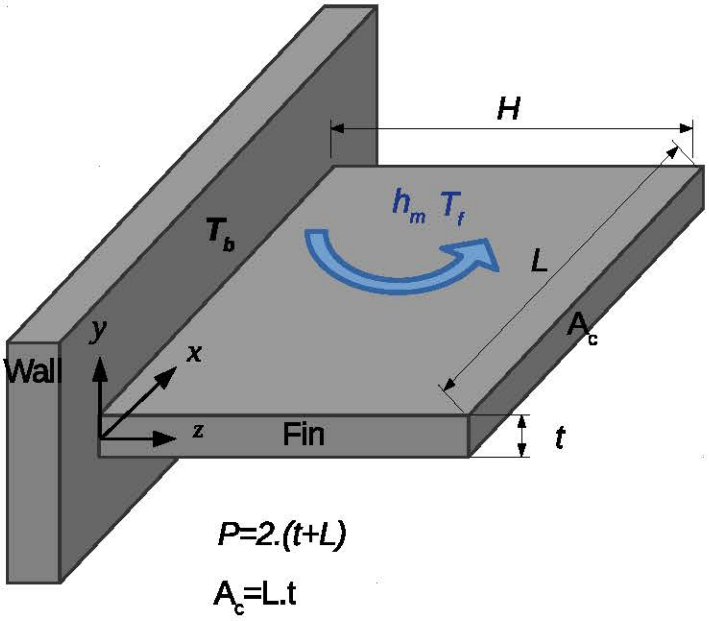


Figure A.3 – One heat fin

The energy balance on a slice of the fin located between z and $z + dz$ leads to the following differential equation [1]:

$$\frac{d^2T}{dz^2} - \frac{hP}{kA_c}(T - T_f) = 0 \tag{A.8}$$

Where A_c is the cross-sectional area, P is the perimeter of the fin cross-sectional area and k is the fin thermal conductivity. The excess temperature θ is defined in order to simplify equation A.8:

$$\theta(z) = T(z) - T_f$$

Equation A.8 becomes

$$\frac{d^2\theta}{dz^2} - m^2\theta = 0 \tag{A.9}$$

Where $m^2 = \frac{hP}{kA_c}$.

The boundary conditions at $z = 0$ is an imposed temperature T_b which can be written for the problem variable as:

$$\theta(z = 0) = T_b - T_f = \theta_b$$

Different boundary conditions at the fin tip ($z = H$) exist [1]. In our case, we consider an adiabatic fin tip.

The solution of the previous problem is:

$$\theta(z) = \theta_b \times \frac{cosh(m.(H-z))}{cosh(m.H)} \tag{A.10}$$

To quantify the temperature gradient within a fin, we define the fin efficiency η_f as the ratio between the real transferred heat and the maximum heat that the fin can transfer (a fin with infinite thermal conductivity):

$$\eta_f = \frac{q_f}{q_{max}} = \frac{q_f}{hA_f\theta_b}$$

Where $A_f = P.H$ is the surface area of the fin and $q_f = \sqrt{h.P.k.A_c}\,\theta_b\, tanh(m.H)$ is the fin heat transfer rate. Thus the fin efficiency can be written as :

$$\eta_f = \frac{tanh(m.H)}{m.H}$$

It is clear that η_f is less or equal than 1. When η_f tends to 1 the fin is isothermal and its performance are improved. A way to calculate the effective heat exchange area is to use the fin's efficiency:

$$A_{eff} = \eta_f\ P.L + \ L.s \tag{A.11}$$

Using this expression in equation A.7 the thermal resistance of a single channel becomes:

$$R_{th}^{channel} = \frac{1}{\dot{m}C_p\left[1 - exp(\frac{-A_eff\ h_m}{\dot{m}C_p})\right]} \tag{A.12}$$

Expression A.12 is in good agreement with the result obtained in [2].

A.0.4 Forced convection correlation

As mentioned in 1.2.1.2 the choice of the convection correlation depends on many factors. Assumptions made in our case are:

1. Laminar flow (in the range of design parameters, the Reynold number is less than the critical one).
2. Internal flow with rectangular duct.
3. Uniform wall temperature.
4. The flow is thermally and hydrodynamically developped.
5. Forced convection.

The convection coefficient is calculated from the Nusselt number. The forced convection correlation that fullfils all the previous points can be written as [3]:

$$Nu_m = 7.55 + \frac{0.024x^{*-1.14}}{1 + 0.0358Pr^{0.17}x^{*-0.64}} \tag{A.13}$$

Where $Pr = \frac{\mu C_p}{k_f}$ is the Prandtl number and $x^* = \frac{x/D_h}{Re\ Pr}$ is the dimensionless axial distance ($x = L$ for this case). Correlation A.13 is valid for $0.1 < Pr < 1000$.

A.0.5 Equivalent heat coefficient of the heat sink

The thermal resistance of a single channel is given by equation A.12 in the case of uniform wall temperature. If we consider N heat fins, then we can see that the heat will cross the $N-1$ channels in parallel as shown in figure A.4.

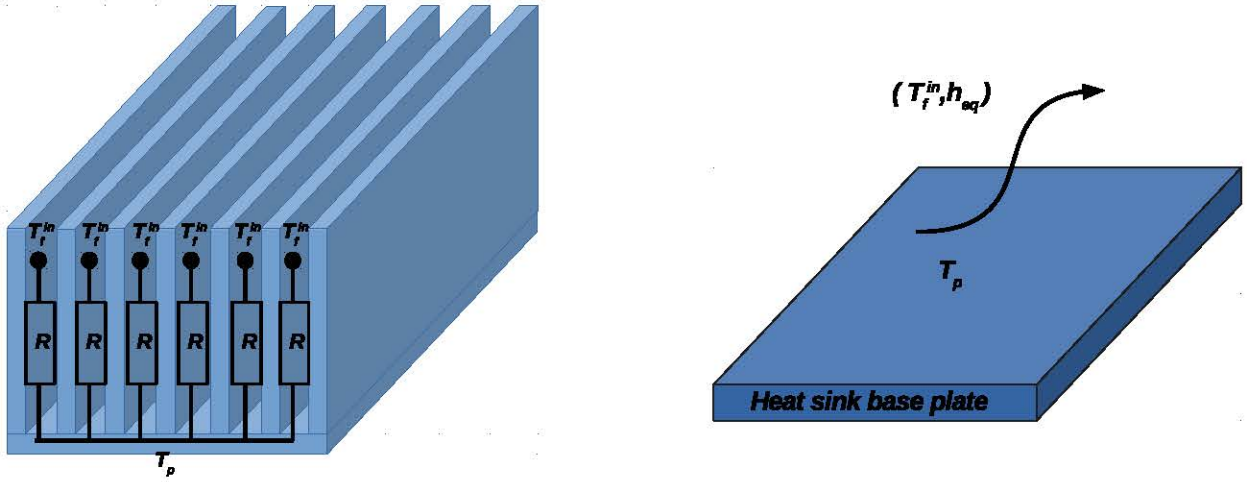


Figure A.4 – (Left) Heat sink thermal network (Right) Equivalent convective boundary condition

Hence, the thermal resistance of the heat sink can be calculated as:

$$\frac{1}{R_{th}^{HS}} = \sum_{i=1}^{N-1} \frac{1}{R_{th}^{channel}} = \frac{N-1}{R_{th}^{channel}} \tag{A.14}$$

As the thermal resistance of all the channels is the same, the thermal resistance of the heat sink R_{th}^{HS} can be obtained by inserting equation A.12 in equation A.14:

$$R_{th}^{HS} = \frac{1}{(N-1)\dot{m}C_p\left[1-\exp(\frac{-A_{eff}\ h_m}{\dot{m}C_p})\right]} \tag{A.15}$$

Let's denote by S_{eq} the upper surface of heat sink base plate. This surface is equal to $S_{eq} = L \times W$, thus the equivalent heat transfer coefficient h_{eq} for the heat sink is defined as :

$$R_{th}^{HS} = \frac{1}{S_{eq}h_{eq}}$$

The heat transfer coefficient can then be obtained as :

$$h_{eq} = \frac{1}{S_{eq}R_{th}^{HS}}$$

This equivalent heat transfer coefficient allows us to model the thermal behavior of the heat sink without building the finite element model. The thermal resistance of the channel is calculated with respect to the inlet fluid temperature thus the overall exchanged heat is equal to:

$$\Phi = h_{eq} S_{eq} \left(T_p - T_f^{in} \right)$$

When performing finite element simulation, the heat sink can be modeled by a simple convective boundary condition, which enables us to gain computing time.

Bibliography

[1] Theodore L Bergman, Frank P Incropera, David P DeWitt, and Adrienne S Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[2] Yuri S Muzychka. Generalized models for laminar developing flows in heat sinks and heat exchangers. In *ASME 2011 9th International Conference on Nanochannels, Microchannels, and Minichannels*, pages 615–628. American Society of Mechanical Engineers, 2011.

[3] Adrian Bejan. *Convection heat transfer*. John wiley & sons, 2013.

Appendix B

Finite Element Method

B.1 Finite element Method for heat transfer in solid medium

B.1.1 Weighted integral formulation

Lets denote by $\hat{T}_s(\mathbf{x},t)$ a trial solution function. The aim of the FEM is to compute $\hat{T}_s(\mathbf{x},t)$. The residual resulting from using the trial function in equations 2.11a is:

$$R(\hat{T}_s(\mathbf{x},t)) = \rho_s\, C_{p,s}\, \frac{\partial \hat{T}_s(\mathbf{x},t)}{\partial t} - \nabla.(k_s\, \nabla \hat{T}_s(\mathbf{x},t)) - Q(\mathbf{x},t) \tag{B.1}$$

The weighted residual method consists in canceling the residual by means of a weight function $T_s^*(\mathbf{x})$ and over the space domain Ω_s. This can be written as :

$$\int_{\Omega_s} R(\hat{T}_s(\mathbf{x},t))\; T_s^*(\mathbf{x})\; d\Omega_s = 0 \tag{B.2}$$

By inserting equation B.1 in equation B.2, the weak formulation becomes:

$$\int_{\Omega_s} \rho_s\, C_{p,s}\, \frac{\partial \hat{T}_s(\mathbf{x},t)}{\partial t}\; T_s^*(\mathbf{x})\; d\Omega_s - \int_{\Omega_s} \nabla.(k_s\, \nabla \hat{T}_s(\mathbf{x},t))\; T_s^*(\mathbf{x})\; d\Omega_s - \int_{\Omega_s} Q(\mathbf{x},t)\; T_s^*(\mathbf{x})\; d\Omega_s = 0 \tag{B.3}$$

The second term of equation B.3 can be integrated using Green theorem (annex C):

$$\int_{\Omega_s} \nabla.(k_s\, \nabla \hat{T}_s(\mathbf{x},t))\; T_s^*(\mathbf{x})\; d\Omega_s = -\int_{\Omega_s} k_s\, \nabla \hat{T}_s.\nabla T_s^*\; d\Omega_s + \int_{\partial\Omega_s} k_s\, \nabla \hat{T}_s\; T_s^*.\mathbf{n}\; d(\partial\Omega_s) \tag{B.4}$$

Therefore, equation B.3 becomes:

$$\begin{aligned}\int_{\Omega_s} \rho_s\, C_{p,s}\, \frac{\partial \hat{T}_s(\mathbf{x},t)}{\partial t}\; T_s^*(\mathbf{x})\; d\Omega_s + \int_{\Omega_s} k_s\, \nabla \hat{T}_s.\nabla T_s^*\; d\Omega_s = \int_{\Omega_s} Q(\mathbf{x},t)\; T_s^*(\mathbf{x})\; d\Omega_s \\ + \int_{\partial\Omega_s} k_s\, \nabla \hat{T}_s\; T_s^*.\mathbf{n}\; d(\partial\Omega_s)\end{aligned} \tag{B.5}$$

Or $\mathbf{q}_n = -k_s \nabla \hat{T}_s.\mathbf{n}$ is the normal conduction heat flux on the solid boundary $\partial\Omega_s = \partial\Omega_{s1} \cup \partial\Omega_{s2}$. Therefore the surface integral of equation B.5 becomes:

$$\int_{\partial\Omega_s} k_s \, \nabla\hat{T}_s \, T_s^*.\mathbf{n} \, d(\partial\Omega_s) = -\int_{\partial\Omega_{s1}} q_n \, T_s^* \, d(\partial\Omega_s) - \int_{\partial\Omega_{s2}} q_n \, T_s^* \, d(\partial\Omega_s) \tag{B.6}$$

Because the boundary $\partial\Omega_{s2}$ is insolated, equation B.6 becomes

$$\int_{\partial\Omega_s} k_s \, \nabla\hat{T}_s \, T_s^*.\mathbf{n} \, d(\partial\Omega_s) = -\int_{\partial\Omega_{s1}} q_n \, T_s^* \, d(\partial\Omega_s) \tag{B.7}$$

Therefore the variational formulation given by equation B.5 takes the form:

$$\int_{\Omega_s} \rho_s \, C_{p,s} \, \frac{\partial\hat{T}_s(\mathbf{x},t)}{\partial t} \, T_s^*(\mathbf{x}) \, d\Omega_s + \int_{\Omega_s} k_s \, \nabla\hat{T}_s.\nabla T_s^* \, d\Omega_s = \int_{\Omega_s} Q(\mathbf{x},t) \, T_s^*(\mathbf{x}) \, d\Omega_s$$
$$- \int_{\partial\Omega_{s1}} q_n \, T_s^* \, d(\partial\Omega_s) \tag{B.8}$$

B.1.2 Meshing and discrete weak formulation

The temperature field $\hat{T}_s(\mathbf{x},t)$ in the system domain can be approximated using shape functions. This approximation is given by equation B.9:

$$\hat{T}_s(\mathbf{x},t) = \sum_{j=1}^{M_{TS}} \chi_j(\mathbf{x}) \, T_{s,j}(t) \tag{B.9}$$

Where $\{\chi_{s,j}(\mathbf{x})\}_{1\leq j\leq M_{TS}}$ are shape functions, $\{T_{s,j}(\mathbf{x})\}_{1\leq j\leq M_{TS}}$ are nodal temperature and M_{TS} is the number of temperature nodes in the solid domain.

Various kind of shape functions can be used : linear, quadratic cubic, etc. Shape functions are usually obtained by Lagrange interpolation. Calculation methods and analytical expresions of these shape functions can be found in [**?**,**?**]. The order of interpolation is related to the number of nodes per element. For instance, figure B.2 represents linear and quadratic shape functions for approximating the temperature in one-dimensional domain discretized into 3 elements. Figure B.1 represents bilinear and quadratic shape functions for approximating the temperature in two-dimensional domain discretized into quadrilateral elements.

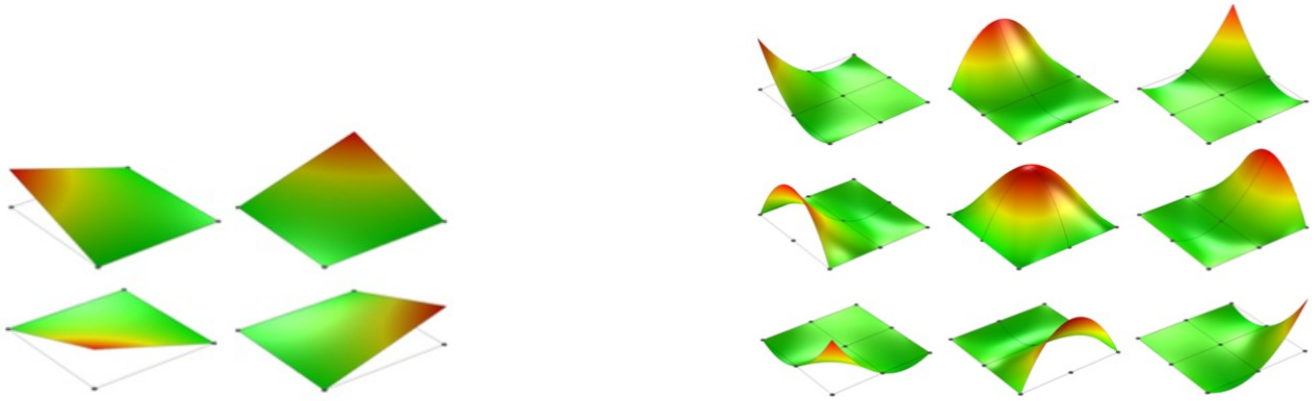


Figure B.1 – Two-dimensional finite element (Left) Bilinear shape functions (Right) Quadratic shape functions

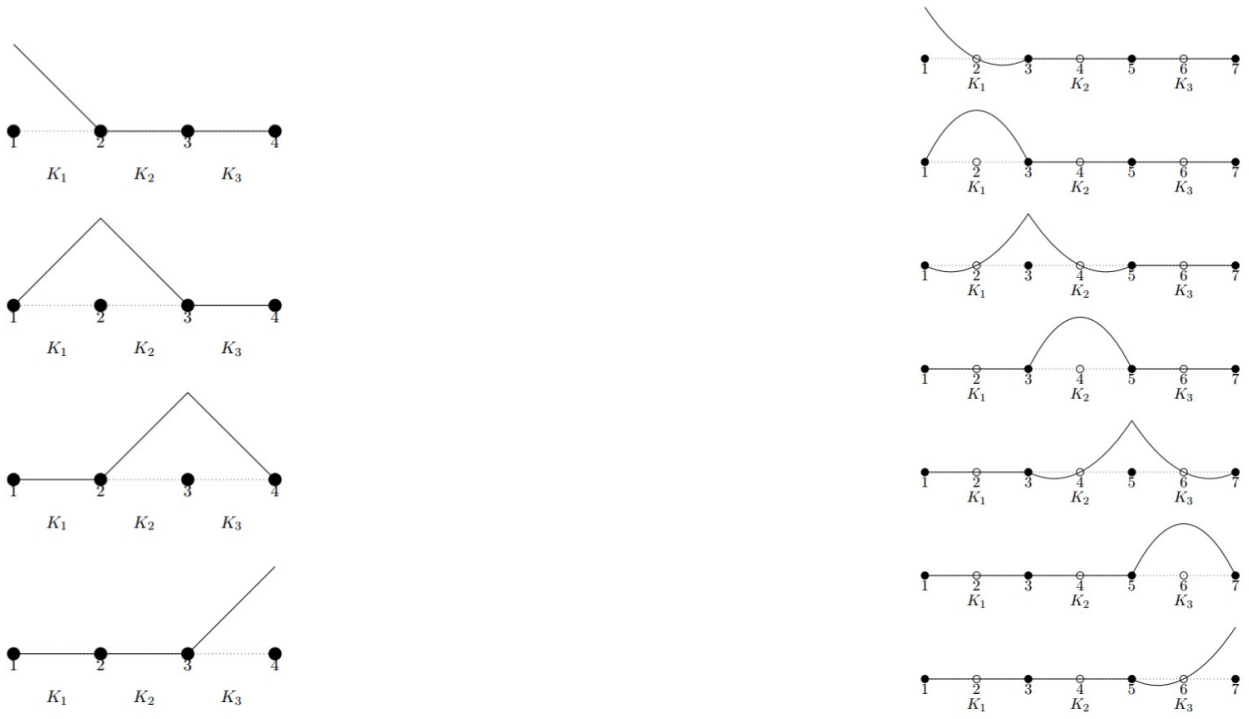


Figure B.2 – One-dimensional finite element (Left) Linear shape functions (Right) Quadratic shape functions

It's obvious that when the number of nodes per element increase (the order increase), the quality of the temperature approximation improves and consequently the temperature is computed with a better accuracy.

Inserting ansatz B.9 in the integral weak formulation yields:

$$\int_{\Omega_s} \rho_s \, C_{p,s} \sum_{j=1}^{M_{TS}} \chi_{s,j}(\mathbf{x}) \frac{dT_{s,j}(t)}{dt} \, T_s^*(\mathbf{x}) \, d\Omega_s + \int_{\Omega_s} k_s \sum_{j=1}^{M_{TS}} \nabla\chi_{s,j}(\mathbf{x}) \, T_{s,j}(t) . \nabla T_s^*(\mathbf{x}) \, d\Omega_s =$$
$$\int_{\Omega_s} Q(\mathbf{x},t) \, T_s^*(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, T_s^*(\mathbf{x}) \, d(\partial\Omega_s) \tag{B.10}$$

Galerking method consists in choosing the weight function identical to the shape function :

$$T_s^*(\mathbf{x}) = \chi_{s,i}(\mathbf{x}) \quad for \;\; i = 1, \ldots, M_{TS} \tag{B.11}$$

The discrete weak formulation given by equation B.10 becomes:

$$\int_{\Omega_s} \rho_s \, C_{p,s} \sum_{j=1}^{M_{TS}} \chi_{s,j}(\mathbf{x}) \frac{dT_{s,j}(t)}{dt} \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s + \int_{\Omega_s} k_s \sum_{j=1}^{M_{TS}} \nabla\chi_{s,j}(\mathbf{x}) . \nabla\chi_{s,i}(\mathbf{x}) \, T_{s,j}(t) \, d\Omega_s =$$
$$\int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, \chi_{s,i}(\mathbf{x}) \;\; d(\partial\Omega_s) \quad for \;\; i = 1, \ldots, M_{TS} \tag{B.12}$$

Interchanging the sum and the integral gives:

$$\sum_{j=1}^{M_{TS}} \left(\int_{\Omega_s} \rho_s \, C_{p,s} \, \chi_{s,j}(\mathbf{x}) \, \chi_{s,i}(\mathbf{x}) \; d\Omega_s \right) \frac{dT_{s,j}(t)}{dt} + \sum_{j=1}^{M_{TS}} \left(\int_{\Omega_s} k_s \, \nabla\chi_{s,j}(\mathbf{x}).\nabla\chi_{s,i}(\mathbf{x}) \, d\Omega_s \right) T_{s,j}(t) =$$
$$\int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, \chi_{s,i}(\mathbf{x}) \, d(\partial\Omega_s) \quad for \;\; i = 1, \ldots, M_{TS} \tag{B.13}$$

Lets denote by :

$$\begin{cases} C_{ij} = \displaystyle\int_{\Omega_s} \rho_s \, C_{p,s} \, \chi_{s,i}(\mathbf{x}) \, \chi_{s,j}(\mathbf{x}) \, d\Omega_s \quad the \;\; mass \;\; matrix & \text{(B.14a)} \\ K_{ij} = \displaystyle\int_{\Omega_s} k_s \, \nabla\chi_{s,i}(\mathbf{x}) \, \nabla\chi_{s,j}(\mathbf{x}) \, d\Omega_s \quad the \;\; stiffness \;\; matrix & \text{(B.14b)} \\ l_i(t) = \displaystyle\int_{\Omega_s} Q(\mathbf{x},t) \, \chi_{s,i}(\mathbf{x}) \, d\Omega_s - \int_{\partial\Omega_{s1}} q_n \, \chi_{s,i}(\mathbf{x}) \; d(\partial\Omega_s) \quad the \;\; load \;\; vector & \text{(B.14c)} \end{cases}$$

The finite element model can be written as :

$$\sum_{j=1}^{M_{TS}} C_{ij} \, \frac{dT_{s,j}(t)}{dt} + \sum_{j=1}^{M_{TS}} K_{ij} \, T_{s,j}(t) = l_i(t) \quad for \;\; i = 1, \ldots, M_{TS} \tag{B.15}$$

Equations B.15 is a $M_{TS} \times M_{TS}$ linear system for the M_{TS} unknown $T_{s,j}$. In matrix form, equation B.15 can be written as:

$$\mathbf{C} \, \frac{d\mathbf{T}_s(t)}{dt} + \mathbf{K} \, \mathbf{T}_s(t) = \mathbf{l}(t) \tag{B.16}$$

B.1.3 Temporal discretization Scheme

Equation B.16 represents an approximation of the PDE 2.11a which is discrete in space and continuous in time. The aim of time integration is to obtain the time evolution of temperature at the mesh nodes. For this purpose, time inetgration method will be applied on equation **??**. The time interval $[0, t_e]$ will be subdevided into N_t time intervals of length $\Delta t = t_{n+1} - t_n = t_n - t_{n-1}$. The outcome of time integration is called the numerical scheme. Many numerical schemes exist and this section will exhibit the most widely used ones.

The most commonly used numerical scheme can be extracted from one-parameter family, called the θ-scheme:

$$\frac{1}{\Delta t}\mathbf{C}(\mathbf{T}_s^{n+1} - \mathbf{T}_s^n) + \theta.\mathbf{K}.\mathbf{T}_s^{n+1} + (1-\theta).\mathbf{K}.\mathbf{T}_s^{n+1} = \theta.\mathbf{l}^{n+1} + (1-\theta).\mathbf{l}^n \tag{B.17}$$

Where θ is a scalar in the interval $[0, 1]$, $\mathbf{T}_s^n$ is the nodal temperature vector at time t_n.
For each value of θ, a well-known numerical scheme is obtained:

- **The forward difference scheme**

  This scheme is also called Forward Euler scheme. It is obtained when $\theta = 0$. Equation B.17 becomes in this case:

$$\frac{1}{\Delta t}\mathbf{C}\mathbf{T}_s^{n+1} = \left(\frac{1}{\Delta t}\mathbf{C} - \mathbf{K}\right)\mathbf{T}_s^n + \mathbf{l}^n \tag{B.18}$$

  For effective implementation of equation B.18, the matrix $\mathbf{C}$ should be invertible. In such case, the numerical scheme presented by equation B.18 can be written as :

$$\mathbf{T}_s^{n+1} = \left(\mathbf{I} - \Delta t\ \mathbf{C}^{-1}\ \mathbf{K}\right).\mathbf{T}_s^n + \left(\Delta t\ \mathbf{C}^{-1}\right).\mathbf{l}^n \tag{B.19}$$

  This scheme is first order accurate in time.

  The main drawback of this scheme is that it is conditionally stable. The stability condition can be written as:

$$\rho\left(\mathbf{I} + \Delta t\ \mathbf{C}^{-1}\ \mathbf{K}\right) \leq 1 \tag{B.20}$$

  Where ρ stands for the spectral radius of a matrix. The stiffness and mass matrix are computed only one time because the considered heat transfer problem is linear. Condition B.20 implies that a critical time step exists beyond which the scheme stability is lost.

- **The Crank-Nicolson scheme**
  This scheme is obtained when $\theta = \frac{1}{2}$ and can be written as :

$$\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)\mathbf{T}_s^{n+1} = \left(\frac{1}{\Delta t}\mathbf{C} - \frac{1}{2}\mathbf{K}\right)\mathbf{T}_s^n + \frac{1}{2}\left(\mathbf{l}^{n+1} + \mathbf{l}^n\right) \tag{B.21}$$

  Once the matrix $\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)$ is invertible, equation B.21 can be written in a practical format:

$$\mathbf{T}_s^{n+1} = \left(\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)^{-1}\left(\frac{1}{\Delta t}\mathbf{C} - \frac{1}{2}\mathbf{K}\right)\right).\mathbf{T}_s^n + \left(\frac{1}{2}\left(\frac{1}{\Delta t}\mathbf{C} + \frac{1}{2}\mathbf{K}\right)^{-1}\right).\left(\mathbf{l}^{n+1} + \mathbf{l}^n\right) \tag{B.22}$$

  This method is inconditionally stable and is second order accurate in time.

- **The backward difference scheme**
  This method is also called Backward Euler method. This scheme is obtained when $\theta = 1$ and can be written as:

$$\left(\frac{1}{\Delta t}\mathbf{C} + \mathbf{K}\right)\mathbf{T}_s^{n+1} = \frac{1}{\Delta t}\mathbf{C}\mathbf{T}_s^n + \mathbf{l}^{n+1} \tag{B.23}$$

  For practical implementation, the matrix $\left(\frac{1}{\Delta t}\mathbf{C} + \mathbf{K}\right)$ should be invertible. When the last requirement is valid, the numerical scheme can be written as :

$$\mathbf{T}_s^{n+1} = \left(\frac{1}{\Delta t}\left(\frac{1}{\Delta t}\mathbf{C} + \mathbf{K}\right)^{-1}\mathbf{C}\right).\mathbf{T}_s^n + \left(\frac{1}{\Delta t}\mathbf{C} + \mathbf{K}\right)^{-1}.\mathbf{l}^{n+1} \tag{B.24}$$

This method is inconditionally stable and is first order accurate in time.

Another well-known time numerical scheme is **the second order Backward Euler method** which can be written as:

$$\frac{1}{2\Delta t}\mathbf{C}\left(\mathbf{T}_s^{n+1}-4\mathbf{T}_s^{n}+\mathbf{T}_s^{n-1}\right)+\mathbf{K}.\mathbf{T}_s^{n+1}=\mathbf{I}^{n+1} \tag{B.25}$$

Equation B.25 can also be arranged as:

$$\left(\frac{3}{2\Delta t}\mathbf{C}+\mathbf{K}\right)\mathbf{T}_s^{n+1}=\frac{2}{\Delta t}\mathbf{C}\mathbf{T}_s^{n}-\frac{1}{2\Delta t}\mathbf{C}\mathbf{T}_s^{n-1}+\mathbf{I}^{n+1} \tag{B.26}$$

Once the matrix $\left(\frac{3}{2\Delta t}\mathbf{C}+\mathbf{K}\right)$ is invertible, the numerical scheme can be written as :

$$\mathbf{T}_s^{n+1}=\left(\frac{2}{\Delta t}\left(\frac{3}{2\Delta t}\mathbf{C}+\mathbf{K}\right)^{-1}\mathbf{C}\right).\mathbf{T}_s^{n}-\left(\frac{1}{2\Delta t}\left(\frac{3}{2\Delta t}\mathbf{C}+\mathbf{K}\right)^{-1}\mathbf{C}\right).\mathbf{T}_s^{n-1}+\left(\frac{3}{2\Delta t}\mathbf{C}+\mathbf{K}\right)^{-1}.\mathbf{I}^{n+1} \tag{B.27}$$

This method is second order accurate in time. When implementing this numerical scheme, the solution at t_n and t_{n-1} are required to compute the solution at t_{n+1}. That's why this scheme is termed as two-step method. In practice, only the initial condition is known. The first time step is performed using a one-step method (for instance Forward or backward Euler scheme or Crank Nicolson scheme).

When performing a multiplication of $M\times M$ matrix by $M\times 1$ vector, $2M^2-M$ elementary operations are carried out. Based on this fact, the number of elementary operation for both backward and forward Euler scheme is $4M^2-M$, for Crank Nicolson scheme it's $4M^2$, while it's equal to $6M^2-M$ for the second order Backwrd Euler method. In order to achieve a proper trade-off between accuracy, low computational expense and easy implementation, the Crank Nicolson scheme will be used in our case.

B.1.4 Finite element Method for heat transfer in fluid medium

In this section, equations 2.6 will be discretized using finite element method.

B.1.4.1 Weighted integral formulation

Using the trial solution $\hat{\mathbf{u}}(\mathbf{x},t)$, $\hat{p}(\mathbf{x},t)$ and $\hat{T_f}(\mathbf{x},t)$ in equations 2.6 yields the following residuals :

$$\begin{cases} R_1(\hat{\mathbf{u}}) = \nabla.(\hat{\mathbf{u}}) & \text{(B.28a)}\\ R_2(\hat{\mathbf{u}},\hat{p},\hat{T_f}) = \rho_0\,\dfrac{\partial\hat{\mathbf{u}}}{\partial t}+\rho_0\,(\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}}+\nabla\hat{p}-\mu\nabla.(\nabla\hat{\mathbf{u}})+\rho_0\,\beta\,(\hat{T_f}-T_0)\,\mathbf{g} & \text{(B.28b)}\\ R_3(\hat{T_f},\hat{\mathbf{u}}) = \rho_0\,C_{p,f}\,\dfrac{\partial\hat{T_f}}{\partial t}+\rho_0.C_{p,f}\,\hat{\mathbf{u}}.\nabla(\hat{T_f})-\nabla.(k_f\,\nabla\hat{T_f}) & \text{(B.28c)}\end{cases}$$

The weighted residual method consists in canceling all the residuals using the weight functions $p^*(\mathbf{x})$, $\mathbf{u}^*(\mathbf{x})$ and $T_f^*(\mathbf{x})$. This can be written as:

$$\begin{cases} \displaystyle\int_{\Omega_f} R_1(\hat{\mathbf{u}}).p^*.d\Omega_f = 0 & \text{(B.29a)} \\ \displaystyle\int_{\Omega_f} R_2(\hat{\mathbf{u}},\hat{p},\hat{T}_f).\mathbf{u}^*.d\Omega_f = 0 & \text{(B.29b)} \\ \displaystyle\int_{\Omega_f} R_3(\hat{T}_f,\hat{\mathbf{u}}).\hat{T_f}^*.d\Omega_f = 0 & \text{(B.29c)} \end{cases}$$

which yields

$$\begin{cases} \displaystyle\int_{\Omega_f} \nabla.(\hat{\mathbf{u}})\ p^*\ d\Omega_f = 0 & \text{(B.30a)} \\ \displaystyle\int_{\Omega_f} \left(\rho_0\ \frac{\partial \hat{\mathbf{u}}}{\partial t}.\mathbf{u}^* + \rho_0\ (\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}}.\mathbf{u}^* + \nabla\hat{p}.\mathbf{u}^* - \mu\ \Delta\hat{\mathbf{u}}.\mathbf{u}^* + \rho_0\ \beta\ (\hat{T}_f - T_0)\ \mathbf{g}.\mathbf{u}^*\right) d\Omega_f = 0 & \text{(B.30b)} \\ \displaystyle\int_{\Omega_f} \rho_0\ C_{p,f}\ \frac{\partial \hat{T}_f}{\partial t}\ T_f^* + \rho_0\ C_{p,f}\ \hat{\mathbf{u}}.\nabla(\hat{T}_f)\ T_f^* - k_f\ \Delta\hat{T}_f\ T_f^*\ d\Omega_f = 0 & \text{(B.30c)} \end{cases}$$

The terms $\displaystyle\int_{\Omega_f} \nabla\hat{p}.\mathbf{u}^*\ d\Omega_f$, $\displaystyle\int_{\Omega_f} \Delta\hat{\mathbf{u}}.\mathbf{u}^*\ d\Omega_f$ and $\displaystyle\int_{\Omega_f} \Delta\hat{T}_f\ T_f^*\ d\Omega_f$ can be developped differently:

development of $\displaystyle\int_{\Omega_f} \nabla\hat{p}.\mathbf{u}^*\ d\Omega_f$

The gradient of a scalar function times a vector can be developped as follows:

$$\nabla(\hat{p}\ \mathbf{u}^*) = \nabla\hat{p}.\mathbf{u}^* + \hat{p}\ \nabla.(\mathbf{u}^*) \tag{B.31}$$

Which yields :

$$\nabla\hat{p}.\mathbf{u}^* = \nabla(\hat{p}\ \mathbf{u}^*) - \hat{p}\ \nabla.(\mathbf{u}^*) \tag{B.32}$$

The gradient theorem allows to compute the integral of $\nabla\hat{p}\ \mathbf{u}^*$ as follows:

$$\int_{\Omega_f} \nabla\hat{p}.\mathbf{u}^*\ d\Omega_f = \int_{\Omega_f} \nabla(\hat{p}\ \mathbf{u}^*)\ d\Omega_f - \int_{\Omega_f} \hat{p}\ \nabla.(\mathbf{u}^*)\ d\Omega_f \tag{B.33}$$

$$= \int_{\partial\Omega_f} \hat{p}\ \mathbf{u}^*.\mathbf{n}\ d(\partial\Omega_f) - \int_{\Omega_f} \hat{p}\ \nabla.(\mathbf{u}^*)\ d\Omega_f \tag{B.34}$$

development of $\displaystyle\int_{\Omega_f} \Delta\hat{\mathbf{u}}.\mathbf{u}^*\ d\Omega_f$

Annex C allows to write :

$$\int_{\Omega_f} \Delta\hat{\mathbf{u}}.\mathbf{u}^*\ d\Omega_f = -\int_{\Omega_f} \nabla\hat{\mathbf{u}}.\nabla\mathbf{u}^*\ d\Omega_f + \int_{\partial\Omega_f} \nabla\hat{\mathbf{u}}.\mathbf{u}^*.\mathbf{n}\ d(\partial\Omega_f) \tag{B.35}$$

development of $\int_{\Omega_f} \Delta\hat{T}_f \; T_f^* \; d\Omega_f$

Annex C allows to write :

$$\int_{\Omega_f} \Delta\hat{T}_f \; T_f^* \; d\Omega_f = -\int_{\Omega_f} \nabla\hat{T}_f.\nabla T_f^* \; d\Omega_f + \int_{\partial\Omega_f} \nabla\hat{T}_f.\mathbf{n} \; T_f^* \; d(\partial\Omega_f) \tag{B.36}$$

By replacing integrals B.34, B.35 and B.36 in equations B.30, the variational formulation becomes:

$$\begin{cases} \displaystyle\int_{\Omega_f} \nabla.(\hat{\mathbf{u}}) \; p^* \; d\Omega_f = 0 & \text{(B.37a)} \\ \displaystyle\int_{\Omega_f} \rho_0 \; \frac{\partial\hat{\mathbf{u}}}{\partial t}.\mathbf{u}^* + \rho_0 \; (\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}}.\mathbf{u}^* \; d\Omega_f + \int_{\partial\Omega_f} \hat{p} \; \mathbf{u}^*.\mathbf{n} \; d(\partial\Omega_f) - \int_{\Omega_f} \hat{p} \; \nabla.(\mathbf{u}^*) \; d\Omega_f & \\ \displaystyle +\mu \int_{\Omega_f} \nabla\hat{\mathbf{u}}.\nabla\mathbf{u}^* \; d\Omega_f - \mu \int_{\partial\Omega_f} \nabla\hat{\mathbf{u}}.\mathbf{u}^*.\mathbf{n} \; d(\partial\Omega_f) + \int_{\Omega_f} \rho_0 \; \beta \; (\hat{T}_f - T_0) \; \mathbf{g}.\mathbf{u}^* \; d\Omega_f = 0 & \text{(B.37b)} \\ \displaystyle\int_{\Omega_f} \rho_0 \; C_{p,f} \; \frac{\partial\hat{T}_f}{\partial t} \; T_f^* + \rho_0 \; C_{p,f} \; \hat{\mathbf{u}}.\nabla(\hat{T}_f) \; T_f^* \; d\Omega_f - k_f \int_{\Omega_f} \nabla\hat{T}_f.\nabla T_f^* \; d\Omega_f & \\ \displaystyle +k_f \int_{\partial\Omega_f} \nabla\hat{T}_f.\mathbf{n} \; T_f^* \; d(\partial\Omega_f) = 0 & \text{(B.37c)} \end{cases}$$

Or

$$\int_{\partial\Omega_f} \hat{p} \; \mathbf{u}^*.\mathbf{n} \; d(\partial\Omega_f) - \mu \int_{\partial\Omega_f} \nabla\hat{\mathbf{u}}.\mathbf{u}^*.\mathbf{n} \; d(\partial\Omega_f) = -\int_{\partial\Omega_f} \boldsymbol{\tau}.\mathbf{u}^*.\mathbf{n} \; d(\partial\Omega_f) \tag{B.38}$$

Where $\boldsymbol{\tau} = \mu \; \nabla\hat{\mathbf{u}} - \hat{p} \; \mathbf{I}$ is the Cauchy stress tensor.

The integral B.38 is canceled by choosing a test function $\mathbf{u}^*(\mathbf{x})$ equal to zero at the boundary $\partial\Omega_f$.

Let's define the heat flux at the fluid boundary as :

$$q^w(\mathbf{x}) = -k_f \; \nabla\hat{T}_f(\mathbf{x}, t).\mathbf{n} \tag{B.39}$$

Thus the variational formulation becomes:

$$\begin{cases} \displaystyle\int_{\Omega_f} \nabla.(\hat{\mathbf{u}}) \; p^* \; d\Omega_f = 0 & \text{(B.40a)} \\ \displaystyle\int_{\Omega_f} \rho_0 \; \frac{\partial\hat{\mathbf{u}}}{\partial t}.\mathbf{u}^* + \rho_0 \; (\hat{\mathbf{u}}.\nabla)\hat{\mathbf{u}}.\mathbf{u}^* \; d\Omega_f - \int_{\Omega_f} \hat{p} \; \nabla.(\mathbf{u}^*) \; d\Omega_f + \mu \int_{\Omega_f} \nabla\hat{\mathbf{u}}.\nabla\mathbf{u}^* \; d\Omega_f & \\ \displaystyle + \int_{\Omega_f} \rho_0 \; \beta \; (\hat{T}_f - T_0)\mathbf{g}.\mathbf{u}^* \; d\Omega_f = 0 & \text{(B.40b)} \\ \displaystyle\int_{\Omega_f} \rho_0 \; C_{p,f} \; \frac{\partial\hat{T}_f}{\partial t} \; T_f^* + \rho_0 \; C_{p,f} \; \hat{\mathbf{u}}.\nabla(\hat{T}_f) \; T_f^* \; d\Omega_f - k_f \int_{\Omega_f} \nabla\hat{T}_f.\nabla T_f^* \; d\Omega_f & \\ \displaystyle + \int_{\partial\Omega_f} q^w(\mathbf{x}) \; T_f^* \; d(\partial\Omega_f) = 0 & \text{(B.40c)} \end{cases}$$

In order to make equation B.43 more compact, the following linear, bilinear and trilinear forms will be defined:

$$\begin{cases} a_u(\hat{\mathbf{u}}, \mathbf{u}^*) = \displaystyle\int_{\Omega_f} \mu \, \nabla\hat{\mathbf{u}}.\nabla\mathbf{u}^* \, d\Omega_f & \text{(B.41a)} \\ a_T(\hat{T}_f, T_f^*) = \displaystyle\int_{\Omega_f} k_f \, \nabla\hat{T}_f.\nabla T_f^* \, d\Omega_f & \text{(B.41b)} \\ a_0(\hat{\mathbf{u}}, p^*) = \displaystyle\int_{\Omega_f} \nabla.(\hat{\mathbf{u}}) \, p^* \, d\Omega_f & \text{(B.41c)} \\ b(\hat{T}_f, \mathbf{u}^*) = \displaystyle\int_{\Omega_f} \rho_0 \, \beta \, (\hat{T}_f - T_0)\mathbf{g}.\mathbf{u}^* \, d\Omega_f & \text{(B.41d)} \\ n_u(\mathbf{u}, \mathbf{v}, \mathbf{u}^*) = \displaystyle\int_{\Omega_f} \rho_0 \, \left(\mathbf{v}.\nabla\right)\mathbf{u}.\mathbf{u}^* \, d\Omega_f & \text{(B.41e)} \\ n_T(\hat{T}_f, \hat{\mathbf{u}}, T_f^*) = \displaystyle\int_{\Omega_f} \rho_0 \, C_{p,f} \, \hat{\mathbf{u}}.\nabla(\hat{T}_f) \, T_f^* \, d\Omega_f & \text{(B.41f)} \\ c(q^w, T_f^*) = \displaystyle\int_{\partial\Omega_f} q^w(\mathbf{x}) \, T_f^* \, d(\partial\Omega_f) & \text{(B.41g)} \\ & \text{(B.41h)} \end{cases}$$

Let's define the following scalar product for two arbitrary vectors **a** and **b** :

$$(\mathbf{a}, \mathbf{b}) = \int_{\Omega_f} \mathbf{a}.\mathbf{b} \, d\Omega_f \tag{B.42}$$

Therefore, the variational formulation given by equations B.43 can be written as:

$$\begin{cases} a_0(\hat{\mathbf{u}}, p^*) = 0 & \text{(B.43a)} \\ \left(\rho_0 \, \dfrac{\partial\hat{\mathbf{u}}}{\partial t}, \mathbf{u}^*\right) + n_u(\hat{\mathbf{u}}, \hat{\mathbf{u}}, \mathbf{u}^*) + a_u(\hat{\mathbf{u}}, \mathbf{u}^*) - a_0(\mathbf{u}^*, \hat{p}) = -b(\hat{T}_f, \mathbf{u}^*) & \text{(B.43b)} \\ \left(\rho_0 \, C_{p,f} \, \dfrac{\partial\hat{T}_f}{\partial t}, T_f^*\right) + n_T(\hat{T}_f, \hat{\mathbf{u}}, T_f^*) + a_T(\hat{T}_f, T_f^*) = -c(q^w, T_f^*) & \text{(B.43c)} \end{cases}$$

The requirement for the existence of the bilinear, trilinear forms used in the weak formulation is that:

- $\hat{p}(\mathbf{x}), \; p^*(\mathbf{x}) \;\in\; H^0(\Omega_f)$
- $\hat{T}_f(\mathbf{x}), \; T_f^*(\mathbf{x}), \; \hat{\mathbf{u}}(\mathbf{x}), \; \mathbf{u}^*(\mathbf{x}) \;\in\; H^1(\Omega_f)$

Where $H^0(\Omega_f)$ and $H^1(\Omega_f)$ are respectively Sobolev space of order 0 and 1.

B.1.4.2 Meshing and discrete weak formulation

The problem unknowns are $\hat{\mathbf{u}}(\mathbf{x},t)$, $\hat{p}(\mathbf{x},t)$ and $\hat{T_f}(\mathbf{x},t)$. These unknowns will be approximated using shape functions:

$$\begin{cases} \hat{\mathbf{u}}(\mathbf{x},t) = \sum_{j=1}^{M_u} \boldsymbol{\sigma_j}(\mathbf{x}).u_j(t) & \text{(B.44a)} \\ \hat{p}(\mathbf{x},t) = \sum_{j=1}^{M_p} \beta_i(\mathbf{x}).p_j(t) & \text{(B.44b)} \\ \hat{T_f}(\mathbf{x},t) = \sum_{j=1}^{M_{Tf}} \chi_{f,j}.T_{f,j}(t) & \text{(B.44c)} \end{cases}$$

Where M_p and M_{Tf} are respectively the number of pressure and temperature nodes in the fluid medium. M_u is the number of velocity degree of freedom. M_u is three times number of velocity nodes when dealing with 3-dimensional fluid dynamic problem.

$u_j(t)$,$p_j(t)$ and $T_{f,j}(t)$ are respectively nodal (at nodes) values of velocity, pressure and temperature at time t. An important rule when choosing shape functions for both velocity and the pressure is that the interpolation used for pressure must be at least one order lower than that used for the velocity field. This rule aims to prevent an overconstrained system of discrete equations [**?**].

Galerkin method consists in substituting the weight functions $\mathbf{u}^*(\mathbf{x})$, $p^*(\mathbf{x})$ and $T_f^*(\mathbf{x})$ by the corresponding shape function $\boldsymbol{\sigma_i}(\mathbf{x})$,$\beta_i(\mathbf{x})$ and $\chi_{f,i}(\mathbf{x})$. $\beta_i(\mathbf{x})$ and $\chi_{f,i}(\mathbf{x})$ are scalar functions whereas $\boldsymbol{\sigma_i}(\mathbf{x})$ is a vector function belonging to $\mathbb{R}^3$ in the case of three dimensional velocity field. The variational formulation can be written as:

$$\begin{cases} a_0(\hat{\mathbf{u}}, \beta_i(\mathbf{x})) = 0 \;\; for \;\; i = 1, \ldots, M_p & \text{(B.45a)} \\ \left(\rho_0.\dfrac{\partial \hat{\mathbf{u}}}{\partial t}, \boldsymbol{\sigma_i}(\mathbf{x})\right) + n_u(\hat{\mathbf{u}}, \hat{\mathbf{u}}, \boldsymbol{\sigma_i}(\mathbf{x})) + a_u(\hat{\mathbf{u}}, \boldsymbol{\sigma_i}(\mathbf{x})) & \\ \qquad -a_0(\boldsymbol{\sigma_i}(\mathbf{x}), \hat{p}) = -b(\hat{T_f}, \boldsymbol{\sigma_i}(\mathbf{x})) \;\; for \;\; i = 1, \ldots, M_u & \text{(B.45b)} \\ \left(\rho_0.C_{p,f}\dfrac{\partial \hat{T_f}}{\partial t}, \chi_{f,i}\right) + n_T(\hat{T_f}, \hat{\mathbf{u}}, \chi_{f,i}) & \\ \qquad +a_T(\hat{T_f}, \chi_{f,i}) = -c(q^w, \chi_{f,i}) \;\; for \;\; i = 1, \ldots, M_{Tf} & \text{(B.45c)} \end{cases}$$

Replacing the finite element approximation of the velocity, pressure and temperature field (equations B.44) in the variational formulation (B.45) yields

$$
\begin{cases}
a_0\left(\sum_{j=1}^{M_u}\boldsymbol{\sigma_j}(\mathbf{x}).u_j(t),\beta_i(\mathbf{x})\right)=0 \;\; for \;\; i=1,\dots,M_p & \text{(B.46a)}\\
\left(\rho_0.\dfrac{\partial\sum_{j=1}^{M_u}\boldsymbol{\sigma_j}(\mathbf{x}).u_j(t)}{\partial t},\boldsymbol{\sigma_i}(\mathbf{x})\right)+n_u\left(\sum_{k=1}^{M_u}\boldsymbol{\sigma_k}(\mathbf{x}).u_k(t),\sum_{j=1}^{M_u}\boldsymbol{\sigma_j}(\mathbf{x}).u_j(t),\boldsymbol{\sigma_i}(\mathbf{x})\right)\\
+a_u\left(\sum_{j=1}^{M_u}\boldsymbol{\sigma_j}(\mathbf{x}).u_j(t),\boldsymbol{\sigma_i}(\mathbf{x})\right)-a_0\left(\boldsymbol{\sigma_i}(\mathbf{x}),\sum_{j=1}^{M_p}\beta_j(\mathbf{x}).p_j(t)\right)\\
=-b\left(\sum_{j=1}^{M_{Tf}}\chi_{f,j}(\mathbf{x}).T_{f,j}(t),\boldsymbol{\sigma_i}(\mathbf{x})\right) \;\; for \;\; i=1,\dots,M_u & \text{(B.46b)}\\
\left(\rho_0.C_{p,f}\dfrac{\partial\sum_{j=1}^{M_{Tf}}\chi_{f,j}(\mathbf{x}).T_{f,j}(t)}{\partial t},\chi_{f,i}(\mathbf{x})\right)+a_T\left(\sum_{j=1}^{M_{Tf}}\chi_{f,j}(\mathbf{x}).T_{f,j}(t),\chi_{f,i}(\mathbf{x})\right)\\
+n_T\left(\sum_{j=1}^{M_{Tf}}\chi_{f,j}(\mathbf{x}).T_{f,j}(t),\sum_{j=1}^{M_u}\boldsymbol{\sigma_j}(\mathbf{x}).u_j(t),\chi_{f,i}(\mathbf{x})\right)\\
=-c\left(q^w,\chi_{f,i}(\mathbf{x})\right) \;\; for \;\; i=1,\dots,M_{Tf} & \text{(B.46c)}
\end{cases}
$$

Linear, bilinear and trilinear forms are linear with respect to each argument (except form $b(\hat{T}_f,\boldsymbol{\sigma_i}(\mathbf{x}))$ which is affine). Thus equations B.46 becomes:

$$
\begin{cases}
\sum_{j=1}^{M_u}a_0\left(\boldsymbol{\sigma_j}(\mathbf{x}),\beta_i(\mathbf{x})\right).u_j(t)=0 \;\; for \;\; i=1,\dots,M_p & \text{(B.47a)}\\
\sum_{j=1}^{M_u}\left(\rho_0.\boldsymbol{\sigma_j}(\mathbf{x}),\boldsymbol{\sigma_i}(\mathbf{x})\right).\dfrac{du_j(t)}{dt}+\sum_{j=1}^{M_u}n_u\left(\sum_{k=1}^{M_u}\boldsymbol{\sigma_k}(\mathbf{x}).u_k(t),\boldsymbol{\sigma_j}(\mathbf{x}),\boldsymbol{\sigma_i}(\mathbf{x})\right).u_j(t)\\
+\sum_{j=1}^{M_u}a_u\left(\boldsymbol{\sigma_j}(\mathbf{x}),\boldsymbol{\sigma_i}(\mathbf{x})\right).u_j(t)-\sum_{j=1}^{M_p}a_0\left(\boldsymbol{\sigma_i}(\mathbf{x}),\beta_j(\mathbf{x})\right).p_j(t)\\
=-\sum_{j=1}^{M_{Tf}}\left(\rho_0\,\beta\,\chi_{f,j}(\mathbf{x})\,\mathbf{g},\boldsymbol{\sigma_i}(\mathbf{x})\right)T_{f,j}(t)+\left(\rho_0\,\beta\,T_0\,\mathbf{g},\boldsymbol{\sigma_i}(\mathbf{x})\right) \;\; for \;\; i=1,\dots,M_u & \text{(B.47b)}\\
\sum_{j=1}^{M_{Tf}}\left(\rho_0.C_{p,f}\chi_{f,j}(\mathbf{x}),\chi_{f,i}(\mathbf{x})\right).\dfrac{dT_{f,j}(t)}{dt}+\sum_{j=1}^{M_{Tf}}a_T\left(\chi_{f,j}(\mathbf{x}),\chi_{f,i}(\mathbf{x})\right).T_{f,j}(t)\\
+\sum_{j=1}^{M_{Tf}}n_T\left(\chi_{f,j}(\mathbf{x}),\sum_{k=1}^{M_u}\boldsymbol{\sigma_k}(\mathbf{x}).u_k(t),\chi_{f,i}(\mathbf{x})\right).T_{f,j}(t)\\
=-c\left(q^w,\chi_{f,i}(\mathbf{x})\right) \;\; for \;\; i=1,\dots,M_{Tf} & \text{(B.47c)}
\end{cases}
$$

Lets define the following matrices

$$\begin{cases} \left(\mathbf{L}_f\right)_{ij} = a_0\left(\boldsymbol{\sigma_j}(\mathbf{x}), \beta_i(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_p \;\; and \;\; 1 \leq j \leq M_u & \text{(B.48a)} \\ \left(\mathbf{C}_{fu}\right)_{ij} = \left(\rho_0 . \boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_u & \text{(B.48b)} \\ \left(\mathbf{K}_{fu}\right)_{ij} = a_u\left(\boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_u & \text{(B.48c)} \\ \left(\mathbf{N}_{fu}\right)_{ij} = n_u\left(\sum_{k=1}^{M_u} \boldsymbol{\sigma_k}(\mathbf{x}).u_k(t), \boldsymbol{\sigma_j}(\mathbf{x}), \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_u & \text{(B.48d)} \\ \left(\mathbf{B}_{fu}\right)_{ij} = \left(\rho_0\, \beta\, \chi_{f,j}(\mathbf{x})\, \mathbf{g}, \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq j \leq M_{Tf} \;\; and \;\; 1 \leq i \leq M_u & \text{(B.48e)} \\ \left(\mathbf{d}\right)_i = \left(\rho_0\, \beta\, T_0\, \mathbf{g}, \boldsymbol{\sigma_i}(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_u & \text{(B.48f)} \\ \left(\mathbf{C}_{fT}\right)_{ij} = \left(\rho_0 . C_{p,f} \chi_{f,j}(\mathbf{x}), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_{Tf} & \text{(B.48g)} \\ \left(\mathbf{K}_{fT}\right)_{ij} = a_T\left(\chi_{f,j}(\mathbf{x}), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_{Tf} & \text{(B.48h)} \\ \left(\mathbf{N}_{fT}\right)_{ij} = n_T\left(\chi_{f,j}(\mathbf{x}), \sum_{k=1}^{M_u} \boldsymbol{\sigma_k}(\mathbf{x}).u_k(t), \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i, j \leq M_{Tf} & \text{(B.48i)} \\ \left(\mathbf{Q}_f\right)_i = \left(q^w, \chi_{f,i}(\mathbf{x})\right) \quad for \;\; 1 \leq i \leq M_{Tf} & \text{(B.48j)} \\ & \text{(B.48k)} \end{cases}$$

Therefore, the weak formulation can then be written as:

$$\begin{cases} \sum_{j=1}^{M_u} \left(\mathbf{L}_f\right)_{ij} . u_j(t) = 0 \quad for \;\; i = 1, \ldots, M_p & \text{(B.49a)} \\ \sum_{j=1}^{M_u} \left(\mathbf{C}_{fu}\right)_{ij} . \frac{du_j(t)}{dt} + \sum_{j=1}^{M_u} \left(\left(\mathbf{K}_{fu}\right)_{ij} + \left(\mathbf{N}_{fu}(\hat{\mathbf{u}})\right)_{ij} \right) . u_j(t) - \sum_{j=1}^{M_p} \left(\mathbf{L}_f\right)_{ij} . p_j(t) & \\ \qquad = - \sum_{j=1}^{M_{Tf}} \left(\mathbf{B}_{fu}\right)_{ij} . T_{f,j}(t) + \mathbf{d}_i \quad for \;\; i = 1, \ldots, M_u & \text{(B.49b)} \\ \sum_{j=1}^{M_{Tf}} \left(\mathbf{C}_{fT}\right)_{ij} . \frac{dT_{f,j}(t)}{dt} + \sum_{j=1}^{M_{Tf}} \left(\left(\mathbf{K}_{fT}\right)_{ij} + \left(\mathbf{N}_{fT}(\hat{\mathbf{u}})\right)_{ij} \right) . T_{f,j}(t) & \\ \qquad = - \left(\mathbf{Q}_f\right)_i \quad for \;\; i = 1, \ldots, M_{Tf} & \text{(B.49c)} \end{cases}$$

Equations B.49 can be cast in the following matrix form:

$$\mathbf{C} . \frac{d\mathbf{X}(t)}{dt} + \mathbf{K}(\mathbf{U}) . \mathbf{X}(t) = \mathbf{L} \tag{B.50}$$

Where

$$\mathbf{X}(t) = \left(\mathbf{P}(t), \mathbf{U}(t), \mathbf{T}_f(t)\right)^T$$

$$\mathbf{U}(t) = \left(u_1(t), u_2(t), \ldots, u_{M_u}(t)\right)$$

$$\mathbf{P}(t) = \big(p_1(t), \ldots, p_{M_p}(t)\big)$$

$$\mathbf{T}_f(t) = \big(T_{f,1}(t), \ldots, T_{f,M_{Tf}}\big)$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{fu} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{fT} \end{pmatrix}$$

$$\mathbf{K}(\mathbf{U}) = \begin{pmatrix} \mathbf{0} & \mathbf{L}_f & \mathbf{0} \\ -\mathbf{L}_f^T & \mathbf{K}_{fu} + \mathbf{N}_{fu}(\mathbf{U}) & \mathbf{B}_f^T \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{fT} + \mathbf{N}_{fT}(\mathbf{U}) \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \\ -\mathbf{Q}_f \end{pmatrix}$$

Appendix C

Green's Formula

Let Ω be a domain in $\mathbb{R}^3$, with boundary $\partial\Omega$, $\mathbf{n}$ is the outward pointing unit normal vector of the boundary $\partial\Omega$ and $\mathbf{f} \in \Omega$.

The divergence theorem have the following form:

$$\int_{\Omega} \nabla.(\mathbf{f})\, d\Omega = \int_{\partial\Omega} \mathbf{f}\, \mathbf{n}\, d(\partial\Omega) \tag{C.1}$$

Setting $\mathbf{f} = \mathbf{f}\, g$, Where g is a scalar function, the left hand side of equation C.1 becomes:

$$\int_{\Omega} \nabla.(\mathbf{f}\, g)\, d\Omega = \int_{\Omega} g\, \nabla.(\mathbf{f})\, d\Omega + \int_{\Omega} \mathbf{f}\, \nabla(g)\, d\Omega \tag{C.2}$$

By replacing the left hand side of equation C.1, the divergence theorem (equation C.1) takes the form :

$$\int_{\Omega} g\, \nabla.(\mathbf{f})\, d\Omega = -\int_{\Omega} \mathbf{f}\, \nabla(g)\, d\Omega + \int_{\partial\Omega} \mathbf{f}\, g\, \mathbf{n}\, d(\partial\Omega) \tag{C.3}$$

Finally, by choosing $\mathbf{f} = \nabla(h)$ we obtain the so-called Green's formula that can be written as:

$$\boxed{\int_{\Omega} g\, \Delta(h)\, d\Omega = -\int_{\Omega} \nabla(h)\, \nabla(g)\, d\Omega + \int_{\partial\Omega} \nabla(h)\, g\, \mathbf{n}\, d(\partial\Omega)} \tag{C.4}$$

Where $\Delta(h) = \nabla.(\nabla(h))$ is the Laplacian of the scalar function h.

Appendix D

Analytical solution of linear first order ODEs

Lets consider the following system of first order ordinary differential equations:

$$\begin{cases} \dfrac{d\mathbf{x}(t)}{dt} = \mathbf{A}\ \mathbf{x}(t) + \mathbf{b}(t) & \text{(D.1a)} \\ \mathbf{x}(t = t_0) = \mathbf{x}_0 & \text{(D.1b)} \end{cases}$$

Where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{b}(t)$ is an $n \times 1$ vector and $\mathbf{x}(t)$ is the unknown vector.
If $\mathbf{A}$ is a diagonalizable matrix, we can write:

$$\mathbf{P}^{-1}.\mathbf{A}.\mathbf{P} = \mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \tag{D.2}$$

Where P is the matrix containing the eigenvectors of $\mathbf{A}$ and $\lambda_1, \cdots, \lambda_n$ are eigenvalues of $\mathbf{A}$.
By replacing $\mathbf{A}$ by its expression from D.2 in equation D.1a we obtain:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{P}.\mathbf{D}.\mathbf{P}^{-1}\ \mathbf{x}(t) + \mathbf{b}(t) \tag{D.3}$$

Right-multiplying equation D.3 by $\mathbf{P}^{-1}$ gives:

$$\mathbf{P}^{-1}\frac{d\mathbf{x}(t)}{dt} = \mathbf{D}.\mathbf{P}^{-1}\ \mathbf{x}(t) + \mathbf{P}^{-1}.\mathbf{b}(t) \tag{D.4}$$

Let's introduce the new state variable $\mathbf{y}(t) = \mathbf{P}^{-1}\mathbf{x}(t)$ then equation D.4 becomes:

$$\frac{d\mathbf{y}}{dt} = \mathbf{D}\ \mathbf{y}(t) + \mathbf{P}^{-1}\ \mathbf{b}(t) \tag{D.5}$$

Equations of D.5 are decoupled, thus each one can be solved like a simple first order differential equation. The solution of D.5 is equal to :

$$\mathbf{y}(t) = \exp(\mathbf{D}\ (t - t_0))\ \mathbf{y}(t_0) + \int_{t_0}^{t} \exp(\mathbf{D}\ (t - \tau))\mathbf{P}^{-1}\mathbf{b}(\tau)\mathrm{d}\tau \tag{D.6}$$

Therefore, the solution of D.1a is given equal to :

$$\boxed{\mathbf{x}(t) = \mathbf{P}\ \exp(\mathbf{D}\ (t - t_0))\ \mathbf{P}^{-1}.\mathbf{x}(t_0) + \mathbf{P}\ \int_{t_0}^{t} \exp(\mathbf{D}\ (t - \tau))\mathbf{P}^{-1}\mathbf{b}(\tau)\mathrm{d}\tau} \tag{D.7}$$

Whe should know that $\exp(\mathbf{D}\ (t - t_0))$ and $\exp(\mathbf{D}\ (t - \tau))$ are diagonal matrix and can be written as :

$$\exp(\mathbf{D}\ (t - t_0)) = \begin{pmatrix} exp(\lambda_1\ (t - t_0)) & 0 & \cdots & 0 \\ 0 & exp(\lambda_2\ (t - t_0)) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & exp(\lambda_n\ (t - t_0)) \end{pmatrix} \tag{D.8}$$

Matrix exponential properties allows to write:

$$\mathbf{P}\ \exp(\mathbf{D}.t)\ \mathbf{P}^{-1} = \exp(\mathbf{P}.\mathbf{D}.\mathbf{P}^{-1}.t) = \exp(\mathbf{A}.t) \tag{D.9}$$

Therefore, solution D.7 can be also be written as:

$$\boxed{\mathbf{x}(t) = \exp(\mathbf{A}\ (t - t_0))\ \mathbf{x}(t_0) + \int_{t_0}^{t} \exp(\mathbf{A}\ (t - \tau))\mathbf{b}(\tau)\mathrm{d}\tau} \tag{D.10}$$

Appendix E

Spatial and temporal modes

E.1 Temporal modes

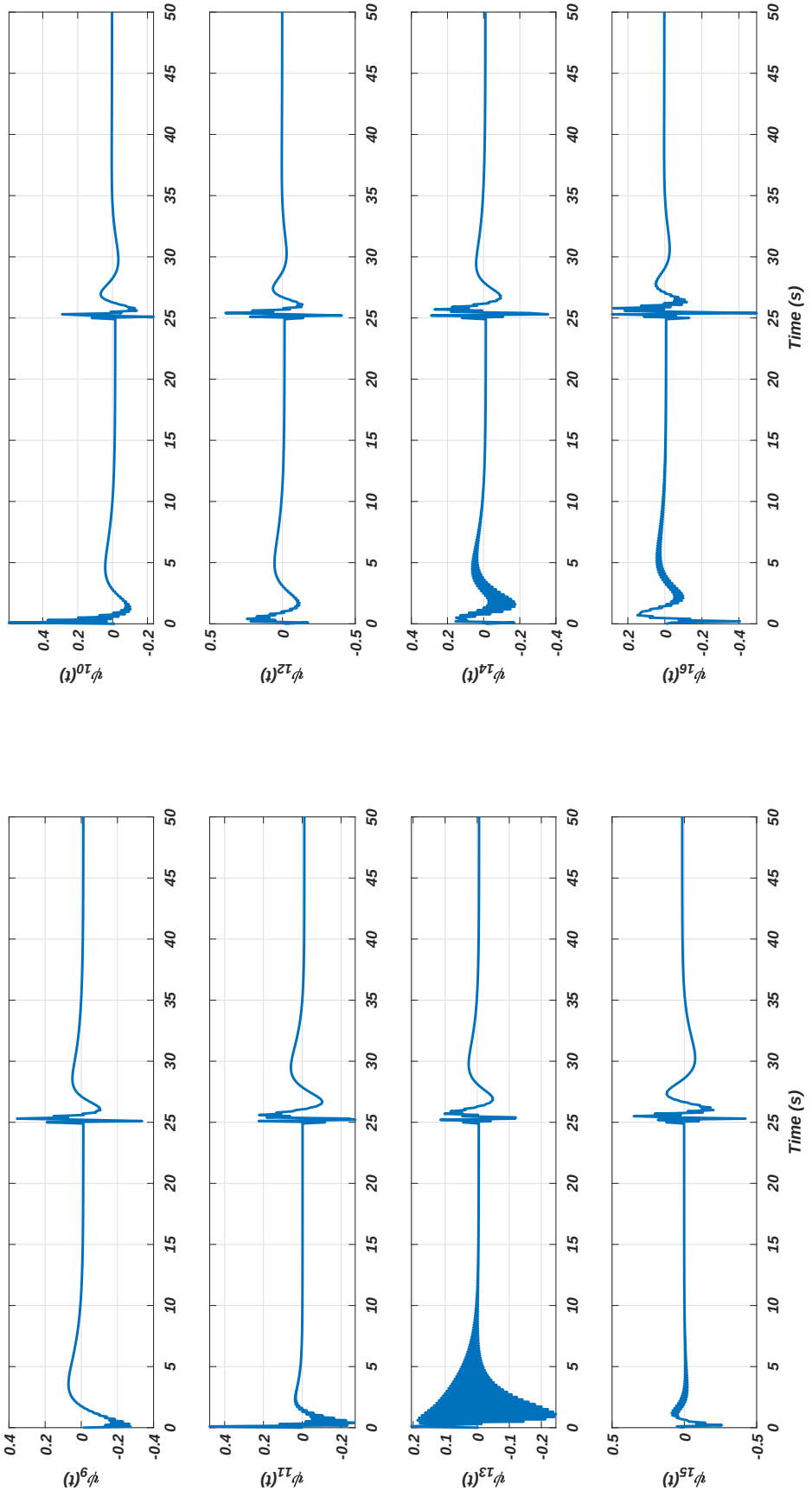


Figure E.1 – The temporal modes of order 9 to 16

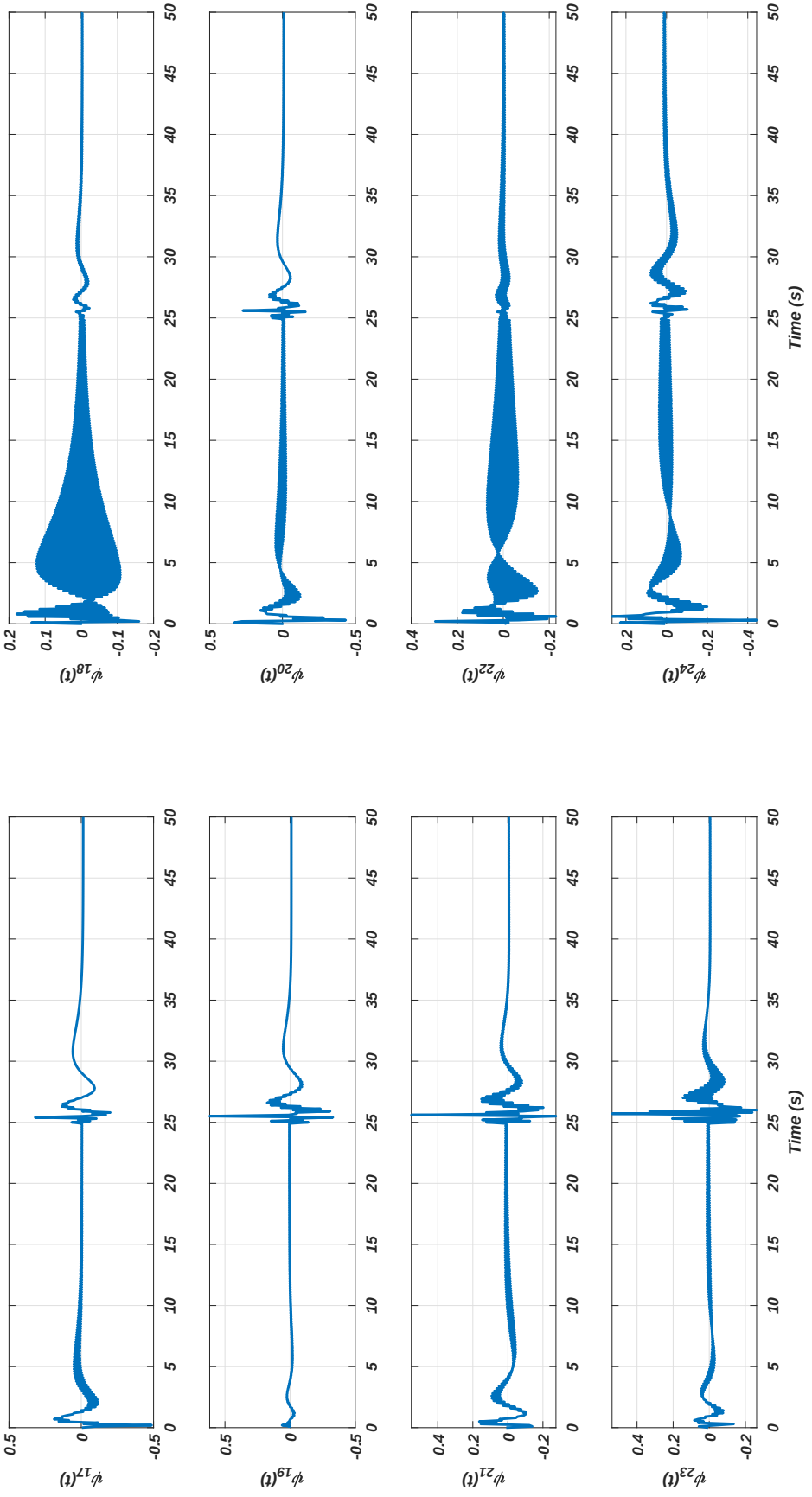
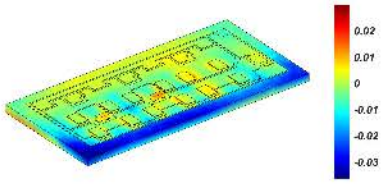
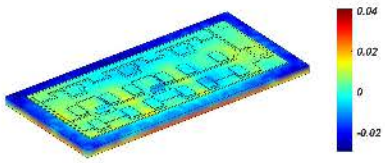


Figure E.2 – The temporal modes of order 17 to 24

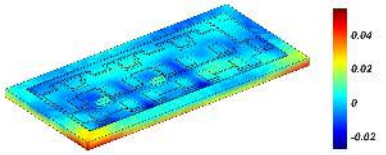
E.2 Spatial Modes



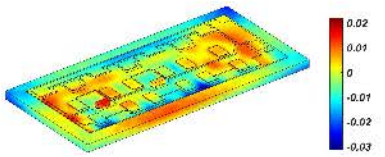
Mode 9



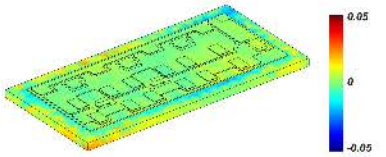
Mode 10



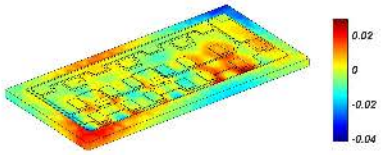
Mode 11



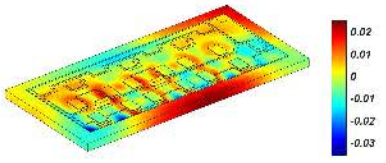
Mode 12



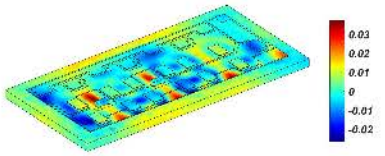
Mode 13



Mode 14

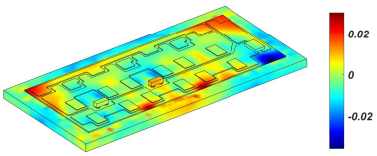


Mode 15

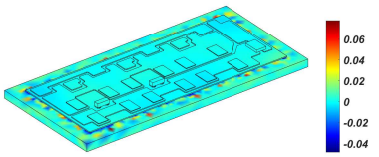


Mode 16

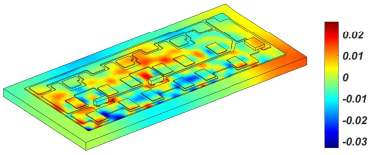
Figure E.3 – The POD modes of order 9 to 16



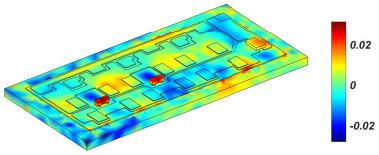
Mode 17



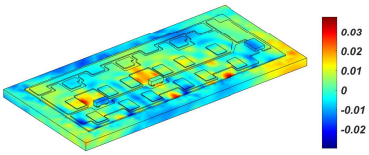
Mode 18



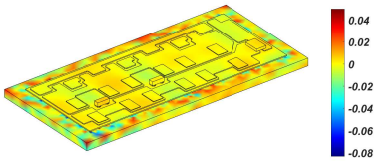
Mode 19



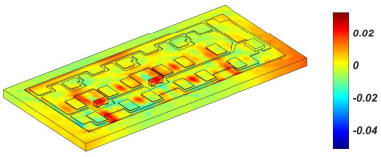
Mode 20



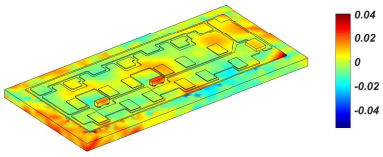
Mode 21



Mode 22



Mode 23



Mode 24

Figure E.4 – The POD modes of order 17 to 24

Appendix F

Additional measurements

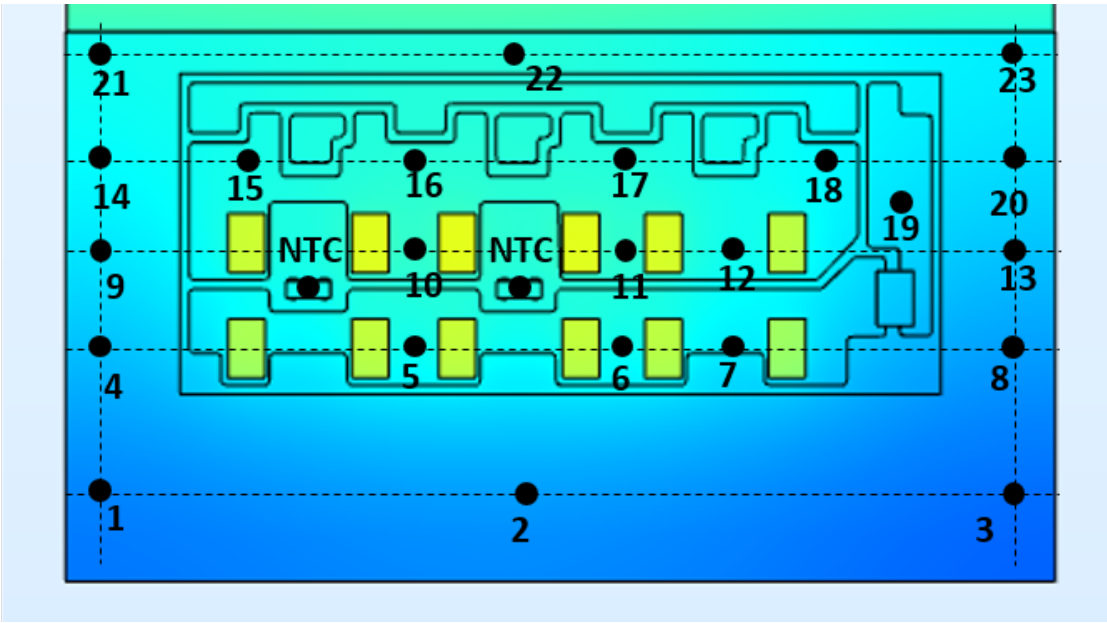


Figure F.1 – Number of measurement locations for one leg

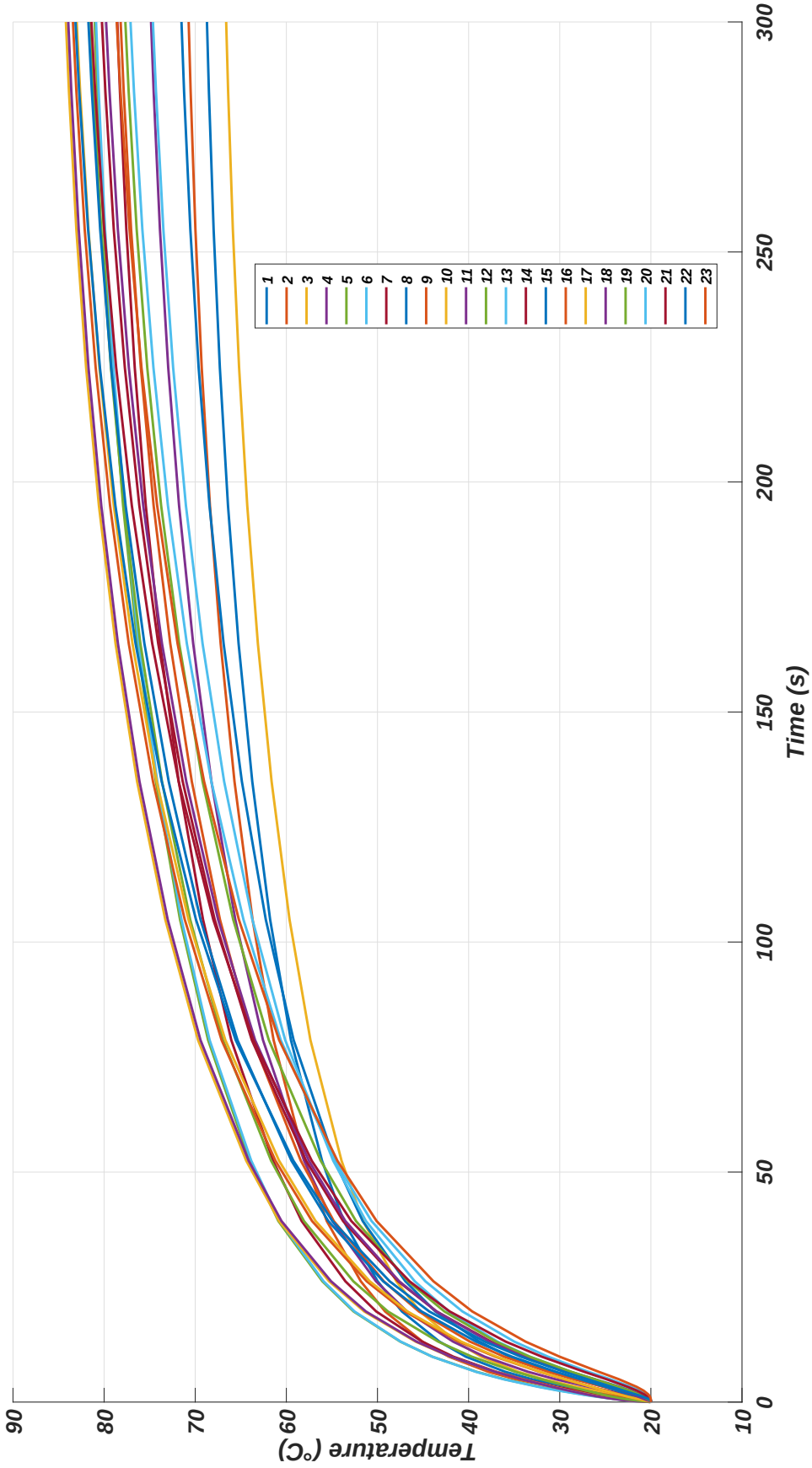


Figure F.2 – The temporal evolution of the temperature in the additional measurement locations

Appendix G

Résumé étendu en Français

G.1 Introduction

L'industrie de transport et en particulier l'aéronautique s'investit de plus en plus dans les systèmes électriques. Cette stratégie se justifie par le cout réduit de maintenance des systèmes électriques vis à vis des commandes hydrauliques. C'est dans ce contexte qu'un consortium européen s'est formé pour construire un convertisseur DC/AC de puissance pour les avions plus électriques. Cette thèse a été réalisée dans le cadre de ce projet européen intitulé I2MPECT dont les membres sont représentés sur la Figure G.1.

Figure G.1 – Consortium européen I2MPECT

Le convertisseur visé par le consortium est construit de transistors de type MOSFET à base de Carbure de Silicium dont la puissance est de $45\,kW$.

La fiabilité des composants électroniques dépend fortement de la température. En effet, $55\,\%$ des défaillances ayant lieu dans les convertisseurs électroniques sont dues aux dépassements de température limite et aux cyclages thermiques (variations temporelles de la température). Les causes de défaillance dans les composants électroniques sont représentées sur la figure G.2.

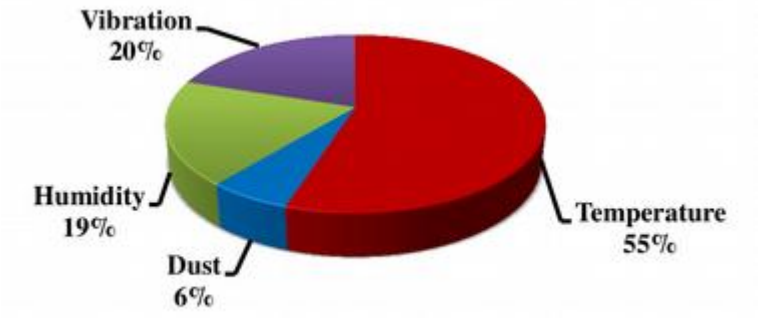


Figure G.2 – Causes de défaillance dans les modules électroniques de puissance [1]

Il est donc nécessaire de surveiller la température des composants électroniques en temps réel afin de suivre leurs états de santé et prédire une éventuelle détérioration au sein du convertisseur. Ceci permet d'appliquer la maintenance préventive dans le convertisseur de puissance.

La mesure directe de la température en plaçant un capteur sur le transistor n'est pas possible pour des raisons d'isolation électrique. Par conséquent, une mesure indirecte est requise, soit par une mesure de grandeurs électriques dépendantes de la température comme la résistance à l'état passant des transistors, soit par une mesure de la température à proximité des transistors (thermistances placées sur la carte électronique), soit par un modèle numérique thermique. Ce modèle de mesure indirecte peut être considéré comme un capteur virtuel. Le modèle mathématique du capteur virtuel doit être à la fois précis et rapide car il doit pouvoir fonctionner sur un micro-contrôleur embarqué. Compte tenu de ces contraintes, un modèle réduit thermique est proposé.

Certaines conditions de fonctionnement pendant lesquelles les ondes électromagnétiques générées par les variations brusques du courant et de tension rendent difficile la mesure de ces deux grandeurs électriques et par conséquent la puissance dissipée calculée à partir du courant est erronée. En plus, l'incertitude sur la résistance électrique à l'état passant des transistors est considérable (La différence relative de la résistance entre deux transistors peut atteindre $25\ \%$). Ces facteurs ont poussé les partenaires à planter 6 capteurs de température (thermistance de type NTC) sur la carte électronique. Ces capteurs étaient localisés légèrement loin des MOSFETs, par conséquent la température mesurée est différente de celle des MOSFETs. L'objectif de ce travail est de construire deux modèles réduits qui répondent à cette problématique :

1. **Un modèle réduit direct :**

   Dans ce modèle les données d'entrée sont la puissance dissipée et les conditions aux limites thermiques du système de refroidissement de la carte électronique. Une représentation schématique de la carte est donnée sur la Figure G.3.

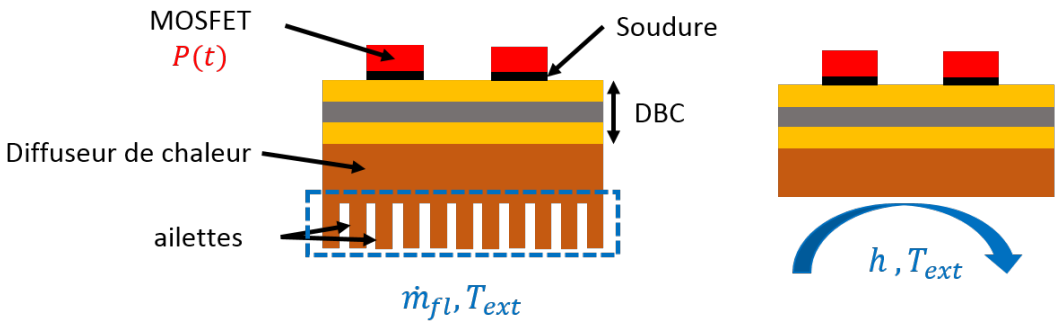


Figure G.3 – Représentation schématique de la carte électronique et de son système de refroidissement

   Ce modèle est obtenu à partir d'un modèle éléments finis et la projection de Galerkin sur une base construite en utilisant la SVD (Singular Value Decomposition). Les détails de cette méthode ainsi que les résultats obtenus seront exposés dans la section G.2.

2. **Un modèle réduit inverse :**

   Dans ce modèle, les seules données d'entrée sont les températures des capteurs placés sur la carte. Ce modèle est construit en utilisant la SVD et un algorithme de minimisation. Les détails de cette méthode ainsi que les résultats obtenus seront exposés dans la section G.3.

G.2 Le modèle réduit direct

G.2.1 Description de la méthode

Une description schématique de cette méthode est donnée sur la Figure G.4 :

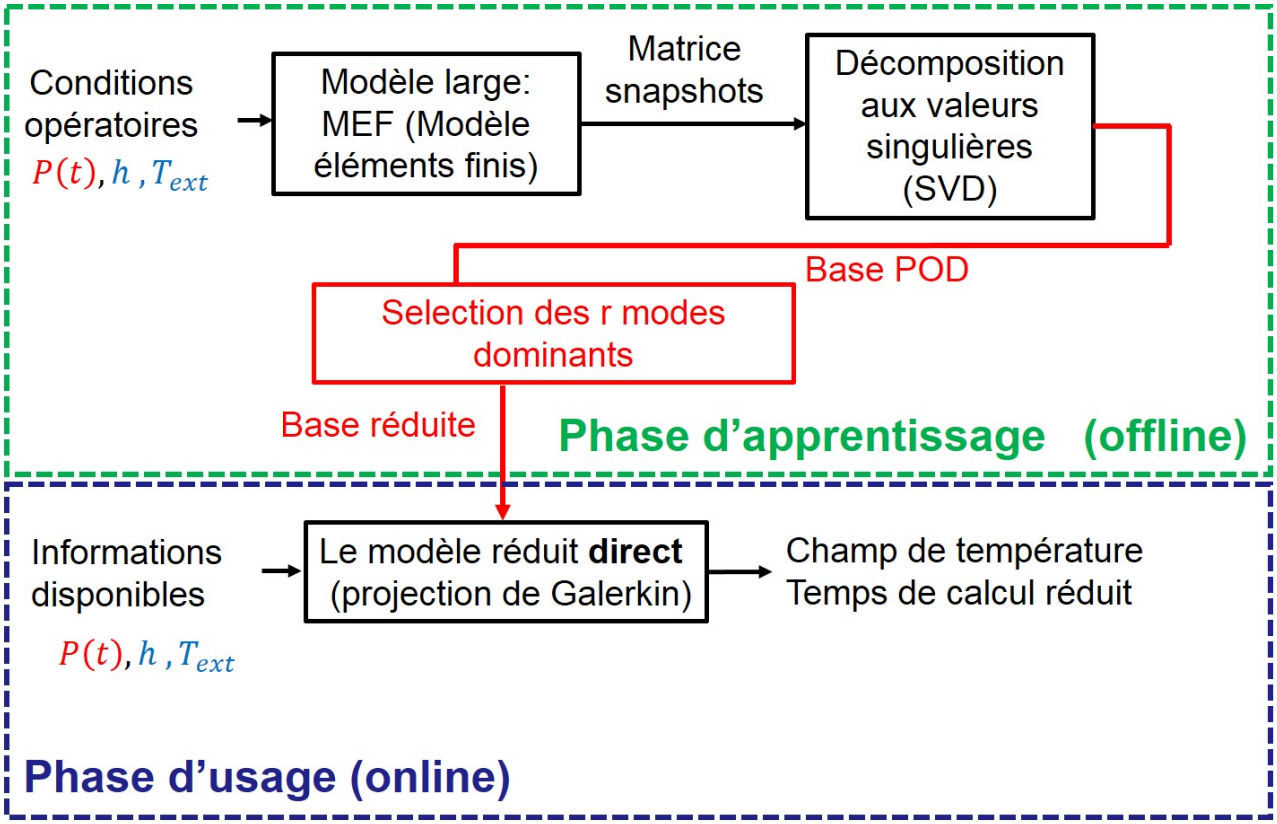


Figure G.4 – Schéma du modèle réduit direct

Cette méthode est basée sur deux phases:

1. **La phase d'apprentissage (offline)**

   Au cours de cette phase, une base réduite qui représente le champ de température est construite. La première étape consiste à construire une matrice nommée snapshots à partir d'une solution calculée par une simulation éléments finis de la carte électronique. Les colonnes de la matrice snapshots sont les champs de température à différents instants. Ensuite, une SVD (Singular Value Decomposition) est appliquée sur la matrice snapshots afin d'extraire une base POD (Proper Orthogonal Decomposition). La contribution des composantes de la base POD ne sont pas de même ordre de grandeur. Une sélection des modes dominants est effectuée en se basant sur la distribution des valeurs singulières afin d'extraire une base réduite. Cette sélection consiste à garder les modes ayant des valeurs singulières supérieures à un seuil.

   Les détails mathématiques de cette phase seront donnés en section G.2.1.1.

2. **La phase d'usage (online)**

   Au cours de cette phase, le champ de température pour un nouveau jeu de conditions opératoires est calculé dans la base réduite. Les coordonnées de la nouvelle solution dans la base réduite sont calculées à partir de la projection de Galerkin du modèle éléments finis sur la base réduite. Cette projection réduit considérablement la taille mathématique du système différentiel à résoudre. Plus de détails sur cette phase seront fournis en section G.2.1.2

G.2.1.1 Construction de la base réduite par POD

Dans cette phase, le champ de température obtenu par le modèle éléments finis est regroupé dans une matrice appelée la matrice des snapshots. Si on note par $\mathbf{u}(t_i)$ le champ de température à l'instant t_i, alors la matrice snapshots $\mathbf{U}$ est définie par l'équation suivante :

$$\mathbf{U} = [\mathbf{u}(t_1), \mathbf{u}(t_2), \ldots, \mathbf{u}(t_N)] \tag{G.1}$$

$\mathbf{U} \in \mathbb{R}^{M \times N}$ avec M représente le nombre des noeuds du modèle éléments finis et N représente le nombre des instants. La SVD est ensuite appliquée sur cette matrice snapshots. Cette décomposition consiste à écrire la matrice snapshots comme suit :

$$\mathbf{U} = \mathbf{\Phi}\ \mathbf{\Sigma}\ \mathbf{\Psi}^T \tag{G.2}$$

avec $\mathbf{\Phi}$, $\mathbf{\Psi}$ sont deux matrices unitaires appartenant respectivement à $\mathbb{R}^{M \times M}$ et $\mathbb{R}^{N \times N}$. $\mathbf{\Sigma}$ est une matrice diagonale contenant les valeurs singulières et appartenant à $\mathbb{R}^{M \times N}$. Par conséquent, le poids de la matrice snapshots est concentré dans la matrice $\mathbf{\Sigma}$. Il a été montré que l'erreur induite par la troncature à l'ordre r de la décomposition SVD peut s'écrire :

$$\|\mathbf{u}(t_i) - \mathbf{u}_r(t_i)\|_\infty \leq \sqrt{E_r} \tag{G.3}$$

avec $\mathbf{u}_r(t_i)$ est le vecteur snapshot reconstruit à partir des r premiers modes.
$E_r = \sqrt{(min(M, N) - r)\ \sigma_{r+1}^2}$ avec σ_i est la i^{me} valeur singulière.
Une caractéristique majeure de la décomposition SVD est la forte décroissance des valeurs singulières. Ceci implique que la solution peut être représentée avec un faible nombre de mode.

Dans la phase suivante le champ solution va être calculé comme combinaison linéaire des r premiers vecteurs propres ($\mathbf{\Phi_r}$), c'est-à-dire:

$$\mathbf{u}(t) \approx \mathbf{\Phi_r}\ \mathbf{a}_r(t) \tag{G.4}$$

avec $\mathbf{a}_r(t)$ est le vecteur contenant les coordonnées de $\mathbf{u}(t)$ dans la base $\mathbf{\Phi_r}$.

G.2.1.2 Projection de Galerkin du modèle éléments finis

Le modèle éléments finis (MEF) de la diffusion de chaleur dans le convertisseur de puissance s'écrit comme suit :

$$\mathbf{C}\ \frac{d\mathbf{u}(t)}{dt} + (\mathbf{K}_0 + h\ \mathbf{K}_{cv})\ \mathbf{u}(t) = P(t)\ \mathbf{l}_Q + h\ T_{ext}\ \mathbf{l}_{cv} \tag{G.5}$$

avec $\mathbf{C}$, $\mathbf{K}_0$ et $\mathbf{K}_{cv}$ sont respectivement la matrice de capacité thermique (ou de masse), les matrices de conductance (ou de raideur). $\mathbf{l}_Q$ et $\mathbf{l}_{cv}$ sont respectivement les vecteurs de charge thermique et de la condition au limite convective. h, T_{ext} et $P(t)$ sont respectivement le coefficient de convection, la température du fluide caloporteur et la puissance totale dissipée aux niveaux des composants électroniques.

La résolution de (G.5) nécessite une condition initiale qui s'écrit :

$$\mathbf{u}(t=0) = \mathbf{u}_0 \tag{G.6}$$

En remplaçant le champ inconnu $\mathbf{u}(t)$ par son expression (G.4) dans le MEF (G.5), on obtient un système différentiel surdéterminé. Pour enlever cette surdétermination, on projette le MEF sur la base réduite. Cette projection est appelée la projection de Galerkin et consiste à multiplier à gauche les équations par ${\boldsymbol{\Phi}_r}^T$. Ces opérations donnent naissance au système suivant:

$$\begin{cases} \mathbf{C}_r \dfrac{d\,\mathbf{a}_r(t)}{dt} + \mathbf{K}_r(h)\,\mathbf{a}_r(t) = \mathbf{l}_r(P(t), h, T_{ext}) & \text{(G.7a)} \\ \mathbf{a}_r(t=0) = \boldsymbol{\Phi}_r^T\,\mathbf{u}_0 & \text{(G.7b)} \\ \mathbf{u}(t) = \boldsymbol{\Phi}_r\,\mathbf{a}_r(t) & \text{(G.7c)} \end{cases}$$

avec

$$\begin{cases} \mathbf{C}_r = \boldsymbol{\Phi}_r^T\,\mathbf{C}\,\boldsymbol{\Phi}_r \ \in \mathbb{R}^{r\times r} & \text{(G.8a)} \\ \mathbf{K}_r(h) = \boldsymbol{\Phi}_r^T\,\mathbf{K}_0\,\boldsymbol{\Phi}_r + h\,\boldsymbol{\Phi}_r^T\,\mathbf{K}_{cv}\,\boldsymbol{\Phi}_r \ \in \mathbb{R}^{r\times r} & \text{(G.8b)} \\ \mathbf{l}_r(P(t), h, T_{ext}) = P(t)\,\boldsymbol{\Phi}_r^T\,\mathbf{l}_Q + h\,T_{ext}\,\boldsymbol{\Phi}_r^T\,\mathbf{l}_{cv} \ \in \mathbb{R}^r & \text{(G.8c)} \end{cases}$$

L'équation (G.7a) compte r équations aux dérivées ordinaires. Comme $r << M$, résoudre l'équation (G.7a) est beaucoup moins couteux que le MEF. Les performances de cette méthode pour calculer le champ de température dans le convertisseur de puissance seront étudiées aux paragraphes suivants.

G.2.2 Résultats et discussions

G.2.2.1 Phase d'apprentissage

La phase d'apprentissage consiste à faire une ou plusieurs simulations éléments finis. Le maillage compte 20000 noeuds. La matrice snapshots a été construite dans le cas $h = 4000\ W/m^2/K$, $T_{ext} = 50\ °C$ et un échelon de puissance représenté sur la Figure G.5.

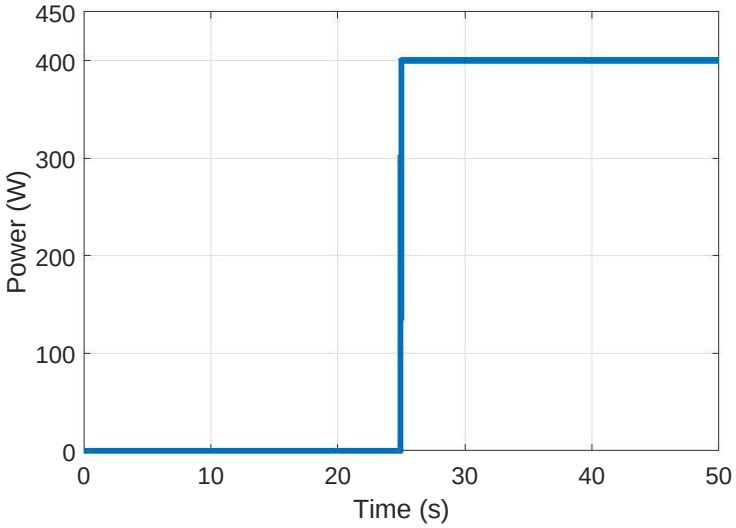
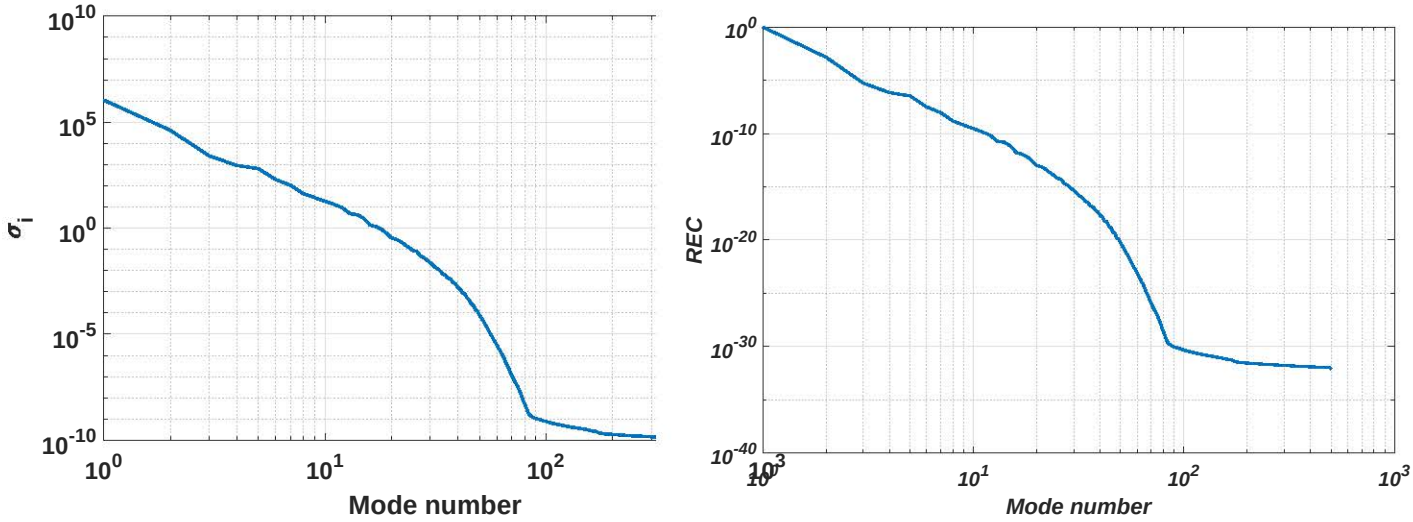


Figure G.5 – Profil de puissance utilisé dans l'apprentissage

La simulation est calculée sur une durée de $50\ s$ avec un pas de temps de $0.1\ s$. Par conséquent, la matrice snapshots appartient à $\mathbb{R}^{20000\times501}$

La distribution des valeurs singulières obtenus par SVD est représentée sur la figure G.6.



Distribution des valeurs singulières Contenu en énergie relatif (REC)

Figure G.6 – Distribution des valeurs singulières et le contenu en énergie relatif (REC)

L'équation (G.3) implique que $r = 25$ modes permettent de reconstruire la température avec une erreur inférieure à $0.1\ °C$.

Si on voulait retenir les modes ayant une énergie relative supérieure à 10^{-10}, alors il faut choisir $r = 10$. Dans la suite, 10 modes seront retenus dans le modèle réduit.

G.2.2.2 Phase d'usage

Maintenant, on considère un profil de puissance différent de celui de l'apprentissage (Figure G.7) tout en conservant les valeurs de h et T_{ext}.

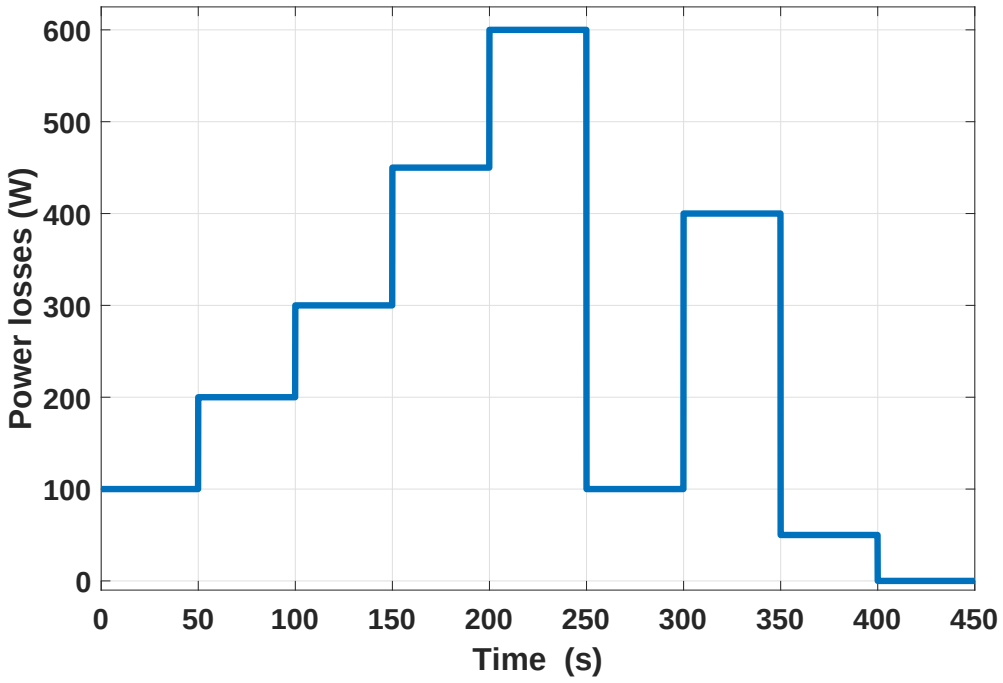
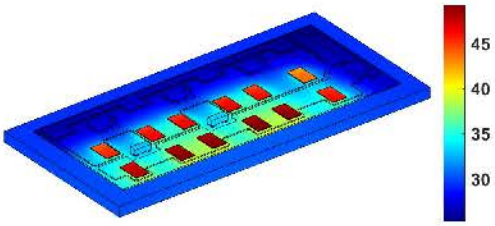
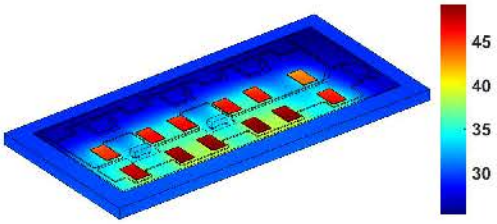


Figure G.7 – Profil de puissance utilisé dans l'usage

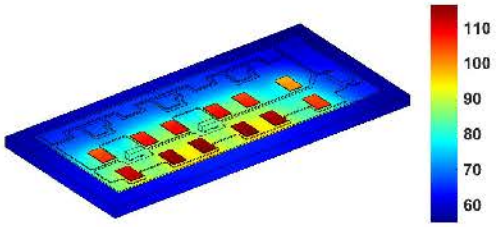
Le champ de température obtenu par le modèle réduit d'ordre 10 et le MEF sont en bon accord comme le montre la Figure G.8.



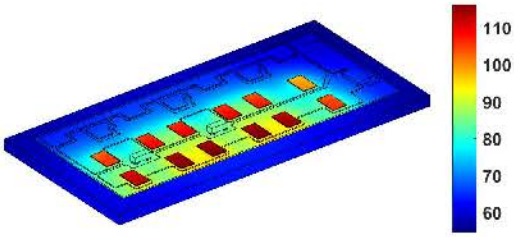
MEF à $t = 1\ s$



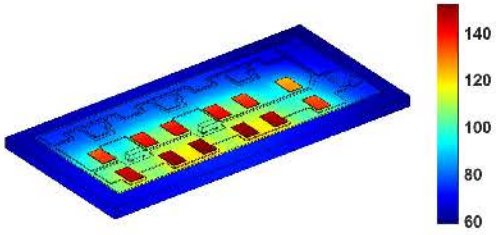
ROM à $t = 1\ s$



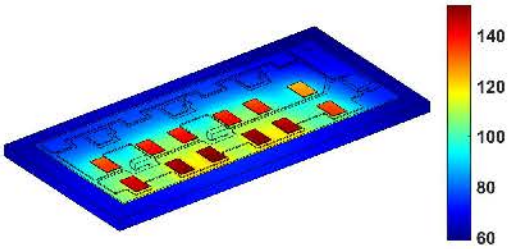
MEF à $t = 55\ s$



ROM à $t = 55\ s$



MEF à $t = 110\ s$



ROM à $t = 110\ s$

Figure G.8 – Champ de température donné par le MEF le ROM aux instants $t = 1\ s$, $t = 55\ s$ et $t = 110\ s$

L'influence du nombre des modes retenus sur la précision du modèle réduit a été étudiée. Sur la Figure G.9, on représente la propagation temporelle de la norme infinie de la différence entre le MEF et le modèle réduit pour plusieurs nombres de modes retenus.

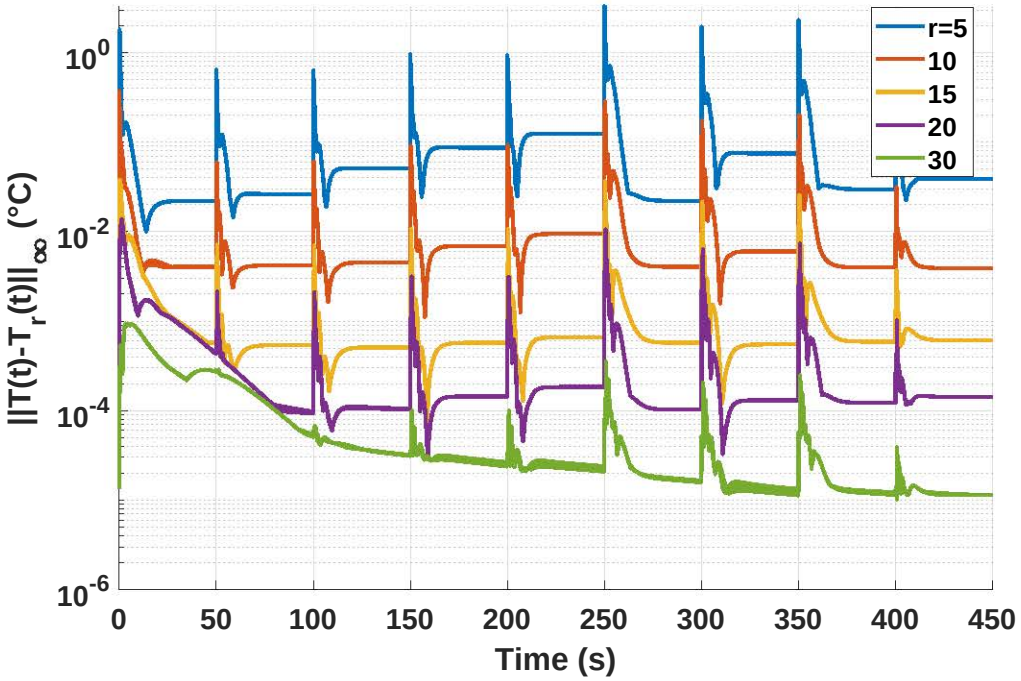


Figure G.9 – Effet du nombre de modes retenus sur la précision du modèle réduit

On constate que l'augmentation du nombre de modes retenus améliore la précision du modèle réduit. Les modes d'ordre supérieur permettent de diminuer les pics d'erreurs pendant les variations brusques de puissances.

Le temps de calcul pour le MEF ainsi que pour les modèles réduits d'ordre différents sont représentés sur le tableau

Model	FEM	ROM (r=10)	ROM (r=20)	ROM (r=30)
CPU Time	11 hours	1.7 seconds	1.8 seconds	1.9 seconds
Gain		23 000	22 000	21000

Table G.1 – Temps de calcul du ROM et du MEF

G.3 Le modèle réduit inverse

Dans le modèle précédent, les paramètres d'entrée du modèle réduit sont les conditions opératoires (h, T_{ext} et $P(t)$). En pratique, il est fort probable que ces données soient indisponibles ou difficiles d'accès pour les raisons mentionnées dans l'introduction. Pour franchir cette barrière, on propose le modèle réduit inverse qui est capable de reconstruire les températures des MOSFETs à partir des températures mesurées par les capteurs de température placés sur la carte.

G.3.1 Description de la méthode

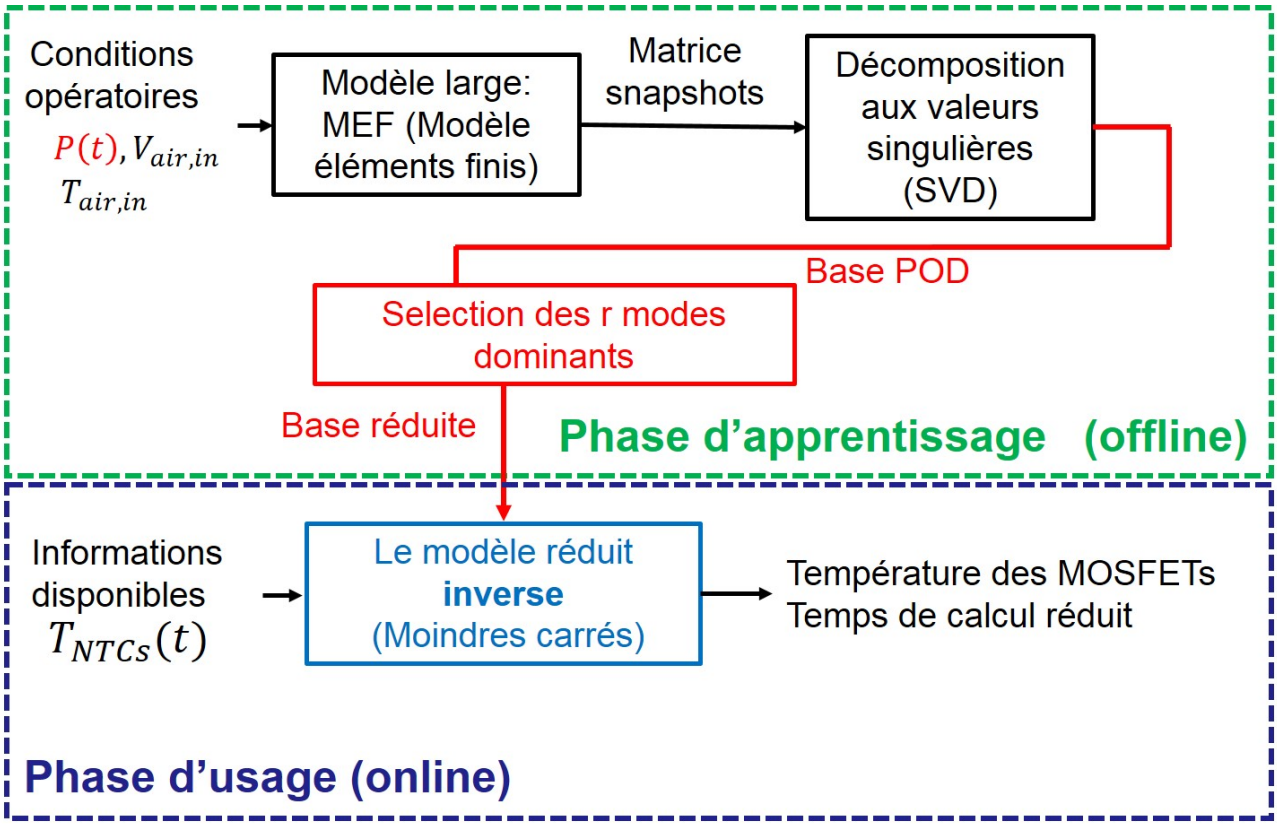


Figure G.10 – Schéma du modèle réduit inverse

Cette méthode est basée sur deux phases :

1. **La phase d'apprentissage (offline)**

   Au cours de cette phase, une base réduite est construite à partir des données fournies par un MEF. On suppose que le champ de température est connu sur tous les nœuds du maillage. La base réduite est obtenue en appliquant une SVD sur la matrice snapshots. La base réduite couple le comportement temporel de la température des MOSFETs et des NTCs. Le formalisme mathématique de cette phase est identique à celui du modèle réduit direct.

2. **La phase d'usage (online)**

   Au cours de cette phase, la température provenant des capteurs de température est exploitée pour calculer la température des MOSFETs. L'idée fondamentale de cette méthode provient du fait que les températures connues (capteurs) et les températures inconnues (MOSFETs) ont les mêmes coordonnées dans la base réduite . Un algorithme de minimisation est appliqué sur les équations correspondantes à l'information connue (capteurs) afin d'identifier les coordonnées dans la base réduite $\mathbf{\Phi}_r$. Le formalisme mathématique sur cette phase sera détaillé dans la suite.

L'objectif de la phase d'apprentissage est d'écrire le champ température dans une base réduite comme indiqué par l'équation suivante :

$$\mathbf{u}_r(t_i) = \sum_{j=1}^{r} a_j(t_i)\ \boldsymbol{\phi}_j = \mathbf{\Phi}_r\ \mathbf{a}_r(t_i) \tag{G.9}$$

La partie connue $\mathbf{u}_{r,kn}(t_i)$ et celle inconnue $\mathbf{u}_{r,unkn}(t_i)$ du champ de température sont données par les équations (G.10) et (G.11) :

$$\mathbf{u}_{r,kn}(t_i) = \mathcal{S}_{kn}.\mathbf{u}_r(t_i) \tag{G.10}$$

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}.\mathbf{u}_r(t_i) \tag{G.11}$$

Avec $\mathcal{S}_{kn}$ et $\mathcal{S}_{unkn}$ sont respectivement la matrice de sélection des températures connues et des températures inconnues.

L'équation (G.9) montre que la partie connue et la partie inconnue ont les mêmes coordonnées dans la base réduite $\mathbf{\Phi}_r$.

Les coordonnées $\mathbf{a}_r(t_i)$ du champ $\mathbf{u}_r(t_i)$ se calculent à partir de la partie connue $\mathbf{u}_{r,kn}(t_i)$.

L'injection de l'équation (G.9) dans les deux équations (G.10) et (G.11) donne :

$$\mathbf{u}_{r,kn}(t_i) = \mathcal{S}_{kn}.\mathbf{\Phi}_r.\mathbf{a}_r(t_i) \tag{G.12}$$

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}.\mathbf{\Phi}_r.\mathbf{a}_r(t_i) \tag{G.13}$$

Les températures connues sont fournies online par les capteurs de températures et seront notées $\mathbf{u}_{r,kn}^{online}(t_i)$. Le seul inconnu dans l'équation (G.12) est le vecteur des coordonnées $\mathbf{a}_r(t_i)$. Pour déterminer $\mathbf{a}_r(t_i)$, le problème de minimisation suivant devra être résolu :

$$\|\mathbf{u}_{r,kn}^{online}(t_i) - \mathcal{S}_{kn}.\mathbf{\Phi}_r.\mathbf{a}_r(t_i)\|_2 \rightarrow min \tag{G.14}$$

Le problème (G.14) contient M_{kn} équations et r inconnus avec $r \neq M$. Par conséquent, un algorithme de minimisation devra être appliqué afin de résoudre (G.14). Une façon possible pour résoudre (G.14) est de calculer la pseudo-inverse de $\mathcal{S}_{kn}.\mathbf{\Phi}_r$ qui sera notée $(\mathcal{S}_{kn}.\mathbf{\Phi}_r)^{\dagger}$. Par conséquent, les coordonnées d'une nouvelle solution dans la base réduite sont :

$$\hat{\mathbf{a}}_r(t_i) = (\mathcal{S}_{kn}.\mathbf{\Phi}_r)^{\dagger}\ .\mathbf{u}_{r,kn}^{online}(t_i) \tag{G.15}$$

La partie inconnue peut être calculée comme suit :

$$\mathbf{u}_{r,unkn}(t_i) = \mathcal{S}_{unkn}.\mathbf{\Phi}_r.\hat{\mathbf{a}}_r(t_i) = \mathcal{S}_{unkn}.\mathbf{\Phi}_r.\left(\mathcal{S}_{kn}.\mathbf{\Phi}_r\right)^{\dagger}.\mathbf{u}^{online}_{r,kn}(t_i) \tag{G.16}$$

On note $\mathbf{A} = \mathcal{S}_{unkn}.\mathbf{\Phi}_r.\left(\mathcal{S}_{kn}.\mathbf{\Phi}_r\right)^{\dagger} \in \boldsymbol{R}^{M_{unkn}\times M_{kn}}$. Alors l'équation (G.16) devient :

$$\mathbf{u}_{r,unkn}(t_i) = \mathbf{A}.\mathbf{u}^{online}_{r,kn}(t_i) \tag{G.17}$$

La matrice $\mathbf{A}$ est calculée une seule fois pour les conditions opératoires d'apprentissage. L'équation G.17 est la plus simple relation mathématique qui peut lier la partie inconnue à la partie connue.

G.3.2 Résultats et discussions

G.3.2.1 Phase d'apprentissage

Durant l'apprentissage, un calcul thermique conjugué (mécanique de fluide et thermique couplés) a été effectué. Le champ de température est calculé par le modèle éléments finis ($COMSOL^{®}$) dans la partie fluide et solide. A titre d'exemple, on représente sur la Figure G.11 la température à l'instant $t = 10s$

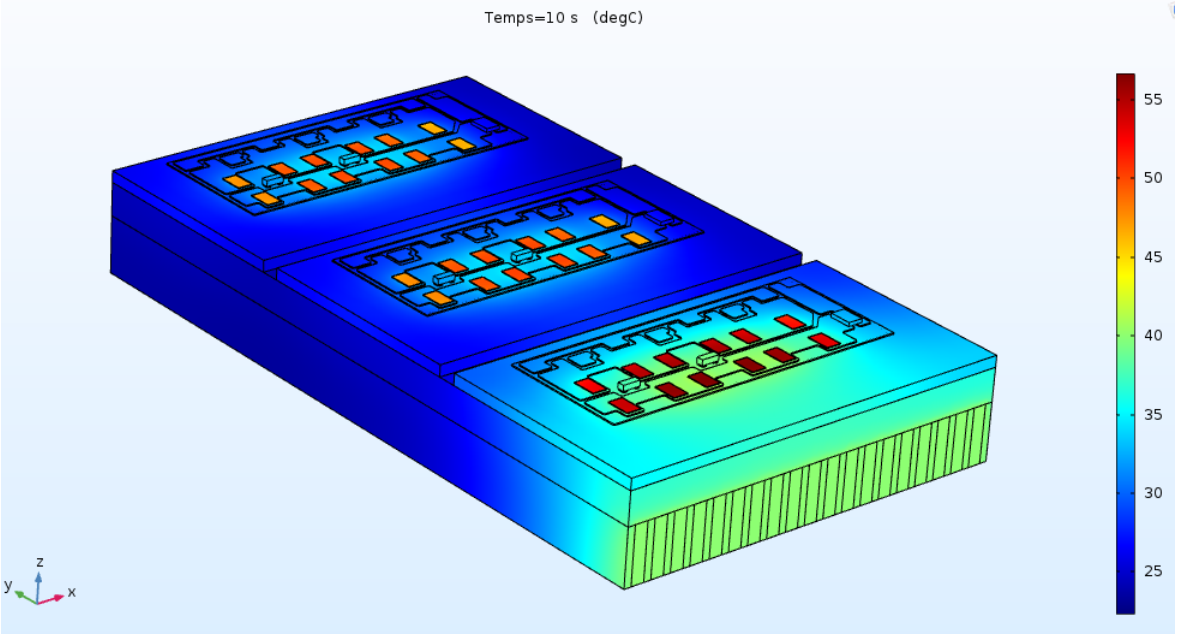


Figure G.11 – Champs de température à $t = 10s$ dans le convertisseur triphasé

Ensuite, on extrait de cette cartographie la température maximale au niveau de chaque MOSFET et la température moyenne de chaque capteur de température. L'emplacement des MOSFETs et des capteurs de température sur la carte électronique est donnée par la Figure G.12 :

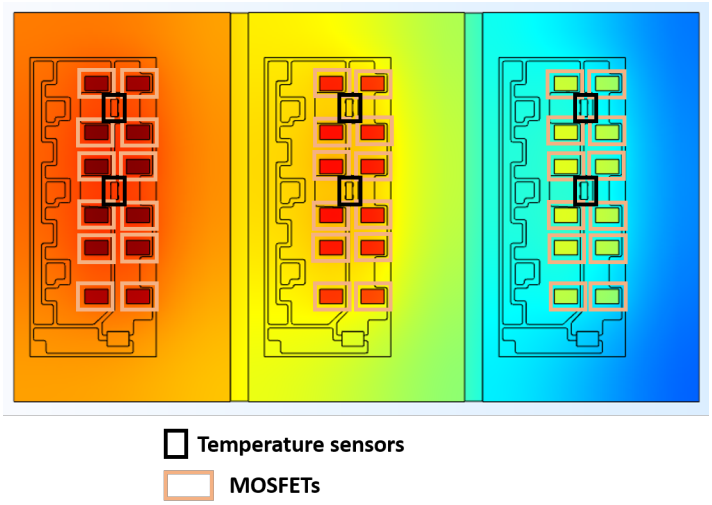
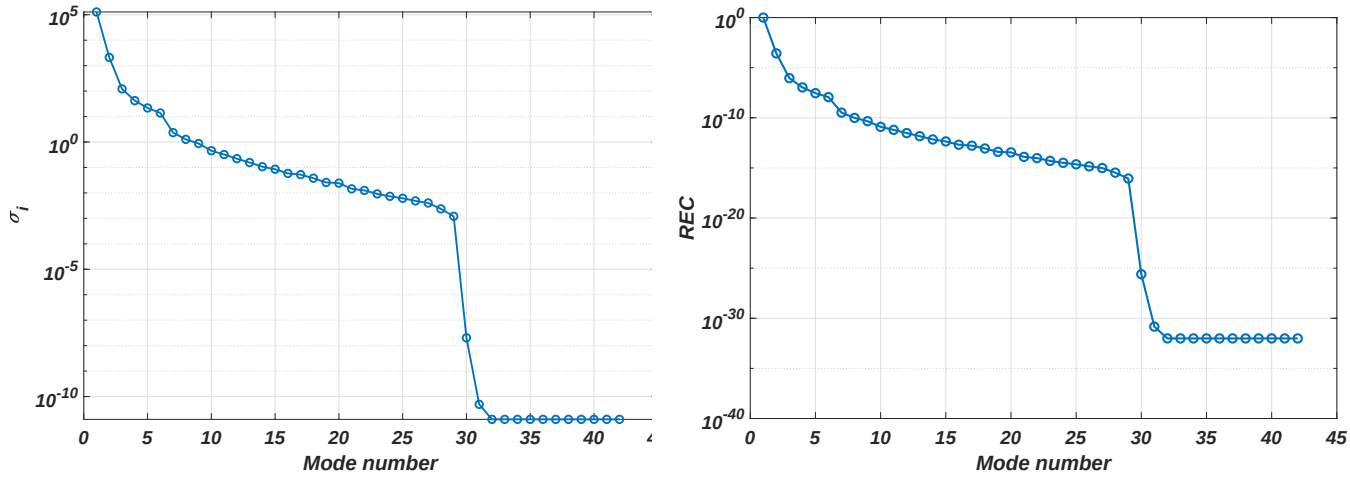


Figure G.12 – Emplacement des MOSFETs et des NTCs

Ces températures constituent la matrice snapshots. La décomposition SVD de la matrice snapshots donne les valeurs singulières présentées sur la Figure G.13 :



Distribution des valeurs singulières Contenu en énergie relatif (REC)

Figure G.13 – Distribution des valeurs singulières et le contenu en énergie relatif (REC)

On constate que les $r = 5$ premiers modes ont une énergie relative supérieure à 10^{-8}. Le spectre des valeurs singulières implique que $r = 15$ modes sont nécessaires pour reconstruire la solution avec une erreur ne dépassant pas $0.1\ °C$.

Une condition importante à respecter lors de la sélection des modes dominants est que le nombre des capteurs doit être supérieur au nombre des modes dominants. C'est une condition nécessaire pour que les coordonnées temporelles identifiées soient uniques. Dans la suite, le modèle réduit inverse sera construit à partir des $r = 5$ premiers modes.

G.3.2.2 Phase d'usage

Dans cette phase, la seule information disponible est les températures données par les capteurs. Par exemple, la Figure G.14 représente l'évolution temporelle de la température dans le MOSFET le plus chaud donnée par les deux modèles : réduits inverse et éléments finis :

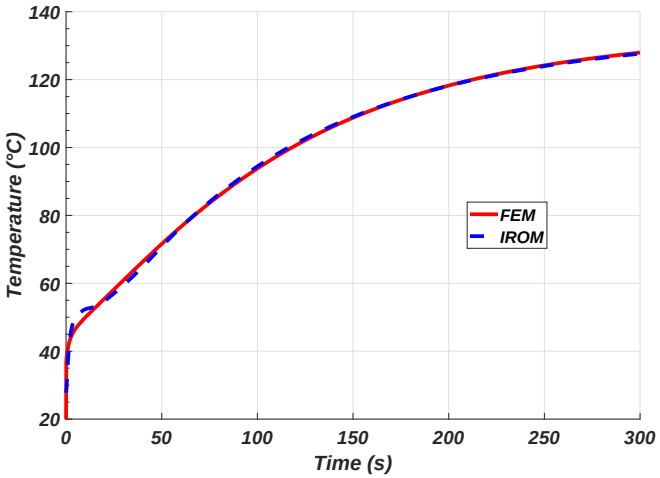


Figure G.14 – Evolution temporelle de la température dans le MOSFET le plus chaud donnée par le MEF et le modèle réduit inverse

La Figure G.14 montre une bonne concordance entre le MEF et le modèle réduit inverse.

L'influence du nombre de modes retenus a été étudiée. Les résultats obtenus (Figure G.15 et G.16) montrent qu'il est obligatoire d'avoir plus de capteurs que de nombre de mode dominant. Dans le cas où le nombre des capteurs est supérieur au nombre de modes retenus, le modèle réduit inverse d'ordre 3 donne la meilleure précision comme représenté sur la Figure G.15.

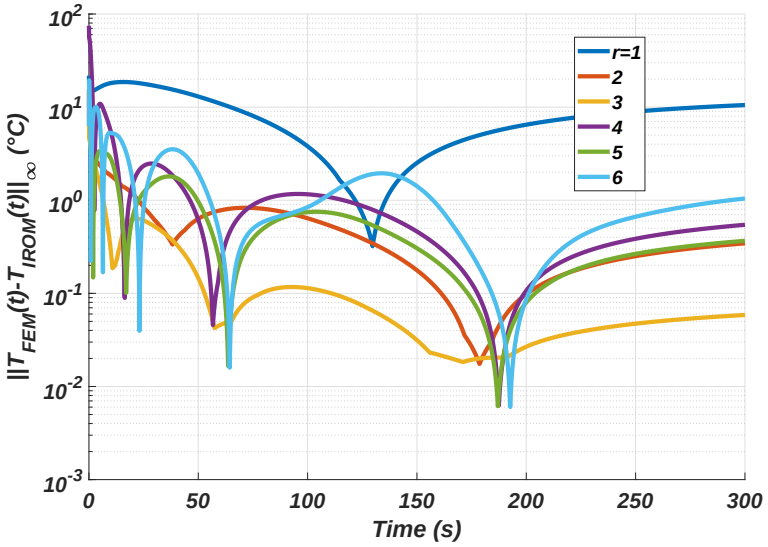


Figure G.15 – Erreur infinie du modèle réduit inverse pour différentes valeurs de modes retenus $r = 1, \ldots, 6$

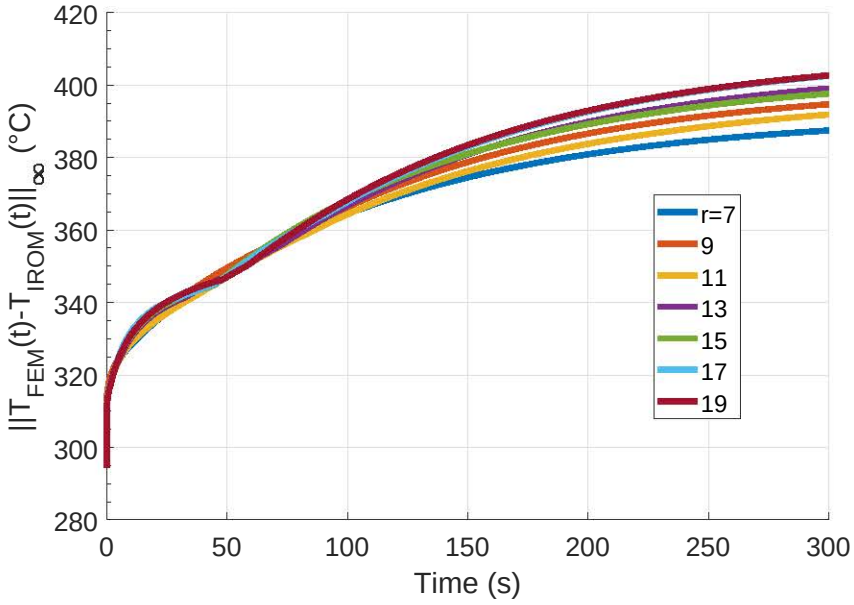


Figure G.16 – Erreur infinie du modèle réduit inverse pour différentes valeurs de modes retenus $r = 7, \ldots, 19$

On s'attendait à ce que le modèle réduit inverse d'ordre $r = 6$ possède la meilleure précision mais ce n'est pas le cas. Ce comportement peut s'expliquer par la dégradation de l'inversibilité de la matrice $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$ lors du processus de minimisation du problème (G.14). En effet, pour que la méthode des moindres carrés fournit une solution unique, il faut que la matrice $(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r)$ soit inversible. Ceci revient à comparer son déterminant à la valeur zéro. Le tableau G.2 donne l'évolution de $det((\mathcal{S}_{kn}\,\mathbf{\Phi}_r)^T\,(\mathcal{S}_{kn}\,\mathbf{\Phi}_r))$ en fonction du nombre des modes retenus.

Number of retained modes r	$det\left((\mathcal{S}_{kn}.\mathbf{\Phi}_r)^T.(\mathcal{S}_{kn}.\mathbf{\Phi}_r)\right)$
1	0.1324
2	0.0187
3	0.0024
4	$1.9\ 10^{-7}$
5	$6\ 10^{-10}$
6	$1.4\ 10^{-15}$
7	$8\ 10^{-28}$
8	$1.6\ 10^{-44}$

Table G.2 – Effet du nombre des modes retenus sur le déterminant de la matrice $(\mathcal{S}_{kn}.\mathbf{\Phi}_r)^T.(\mathcal{S}_{kn}.\mathbf{\Phi}_r)$

On constate que le déterminant diminue quand le nombre des modes augmente et qu'au delà de $r = 4$, le déterminant devient proche des erreurs d'arrondi.

Dans la suite, on va considérer un modèle réduit inverse d'ordre $r = 3$ et on choisit des conditions opératoires différentes de ceux de l'apprentissage. On fait varier respectivement la température du fluide caloporteur, sa vitesse et la puissance dissipée au niveau des MOSFETs. Les jeux de données utilisés dans l'usage et l'apprentissage sont représentés sur la Figure G.17.

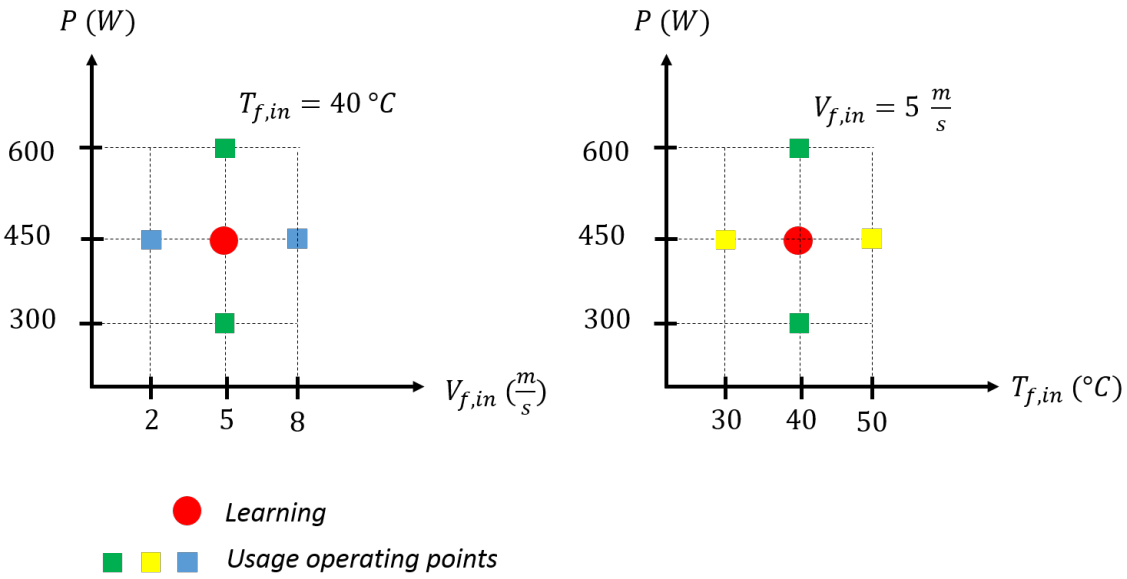


Figure G.17 – Paramètres d'usage et d'apprentissage

Pour chaque cas, la norme infinie de la différence entre le modèle réduit inverse et le MEF est représentée sur les Figures G.18, G.19 et G.20.

On constate que changer les conditions d'usage par rapport à l'apprentissage génère une erreur. Cette erreur est plus prononcée quand on change la température d'entrée d'air de refroidissement ou la puissance dissipée dans les MOSFETs. La Figure G.19 montre que l'erreur du modèle réduit n'est pas sensible au changement de la vitesse d'entrée de l'air. Dans les trois cas, l'erreur calculée est inférieur à $0.6\ °C$ (en dehors des changement brusques de la puissance dissipée).

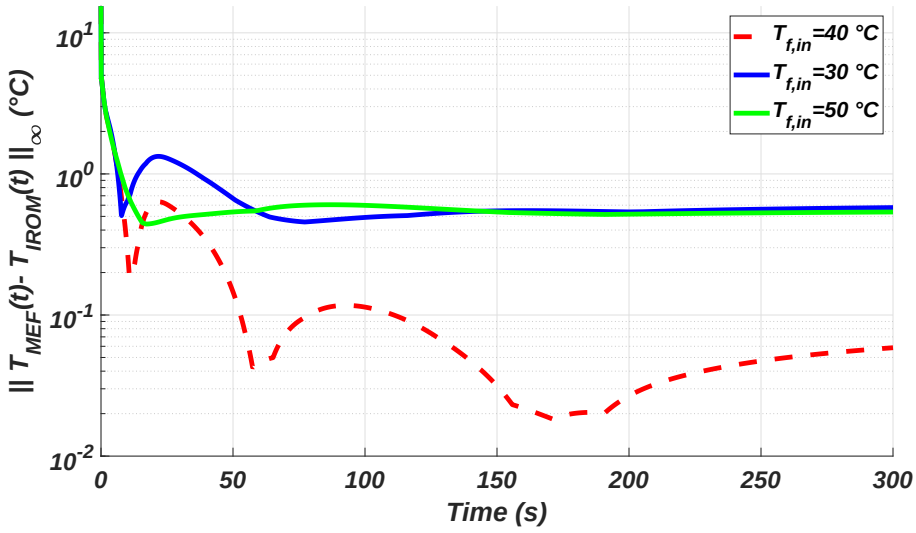


Figure G.18 – Erreur du modèle réduit inverse suite à un changement de la température d'entrée d'air

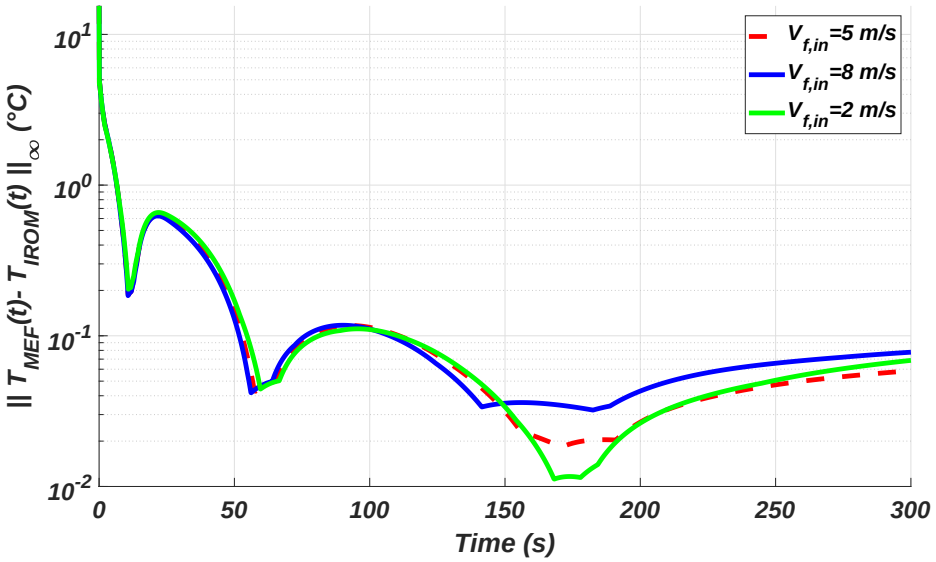


Figure G.19 – Erreur du modèle réduit inverse suite à un changement de la vitesse d'entrée d'air

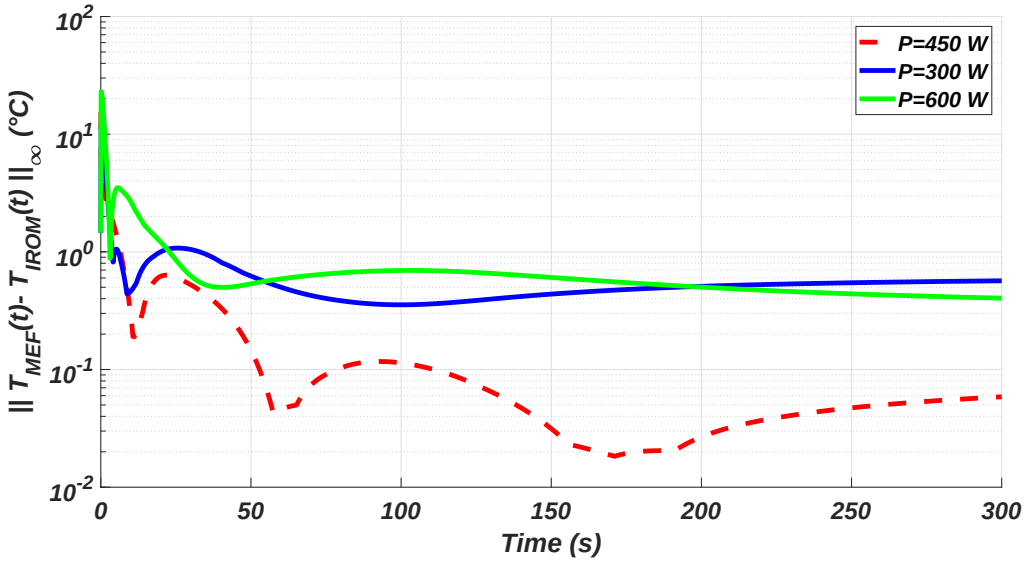


Figure G.20 – Erreur du modèle réduit inverse suite à un changement de la puissance dissipée

G.4 Conclusions et perspectives

Dans ce travail, on a pu construire deux modèles réduits thermiques pour la carte électronique dont la différence réside dans le type de paramètres d'entrées.

Le premier modèle réduit est direct car il a besoin de connaitre les conditions aux limites pour calculer le champ de température. Grâce à la projection de Galerkin et la SVD, le modèle éléments finis (MEF) a pu être réduit toute en préservant sa précision. Les simulations thermiques faites sur une phase du module de puissance montrent que l'erreur du modèle réduit d'ordre 10 est inférieure à $0.01\ ^{\circ}C$ (en dehors des transitions brusques de la puissance dissipée). Les pics de l'erreur durant les variations brusques de puissance sont amortis avec l'ajout des modes.

Le deuxième modèle réduit est inverse car les conditions aux limites ne sont pas nécessaires pour calculer les températures des MOSFETs. Cependant, le modèle a besoin de connaitre les températures des capteurs placés sur la carte électronique. Grâce à la SVD, une base réduite couplant le comportement temporel entre les températures connues (température des capteurs) et les températures inconnues (température des MOSFETs) est construite. Les coordonnées d'une nouvelle solution sont calculées à partir des températures connues mesurées online. Les résultats obtenus montrent une bonne concordance entre le modèle de référence (MEF) et ceux obtenus avec le modèle réduit inverse. Une condition nécessaire pour la réussite de cette approche est que le nombre de capteurs devra être supérieur au nombre de modes retenus. L'effet du changement des conditions d'usage par rapport à l'apprentissage a été étudié. Les résultats obtenus montrent que la précision du modèle réduit inverse est affecté par le changement de la température d'entrée d'air de refroidissement ainsi que de la puissance dissipée. La vitesse d'entrée de l'air de refroidissement a un effet négligeable sur la précision du modèle réduit inverse. Dans tous les cas, l'erreur absolue du modèle est inférieure à $0.6\ ^{\circ}C$ (en dehors des transitions brusques de la puissance dissipée).

Bibliography

[1] Michael Pecht. *Handbook of electronic package design*, volume 76. CRC Press, 1991.

Résumé

Dans la transition vers l'avion plus électrique, un des verrous technologiques est l'échauffement des composants électroniques ce qui affecte fortement leurs fiabilités et leurs durées de vie. En conséquence, il est nécessaire de contrôler la température des cartes électroniques. Cette thèse a pour objectif de construire deux modèles thermiques réduits permettant de suivre en temps réel la température des composants électroniques. La différence entre le modèle réduit directe (MRD) et le modèle réduit inverse (MRI) réside dans les paramètres d'entrée et le formalisme mathématique de leurs constructions.

Pour le MRD, les paramètres d'entrées sont les conditions aux limites. Ce modèle est élaboré sur deux étapes. La première étape consiste à construire une base réduite représentative de la solution en appliquant la POD (Proper Orthogonal Decomposition) sur une matrice snapshots. La matrice snapshots regroupe la solution du modèle éléments finis (MEF). La deuxième étape consiste à calculer les coordonnées d'une nouvelle solution en utilisant la projection de Galerkin du MEF sur la base réduite. Un MRD construit avec 10 modes diminue considérablement le temps de calcul et réalise une erreur absolue inférieure à $0,1\,°C$ en dehors des variations brusques de puissance.

Pour le MRI, les paramètres d'entrée sont les températures des capteurs implantés loin des composants électroniques. Ce modèle n'a pas besoin de connaitre les conditions aux limites comme le MRD. La première étape consiste à construire une base réduite qui couple les températures dans les composants électroniques et dans les capteurs de températures en utilisant la POD. La deuxième étape consiste à identifier les coordonnées des températures des composants électroniques à partir des mesures en utilisant un algorithme de minimisation. L'erreur du MRI d'ordre 3 ne dépasse pas $0,6\,°C$ en dehors des variations brusques de puissance.

Mots clés : Réduction de modèle, POD (Proper Orthogonal Decomposition), thermique des composants électroniques.

Abstract

In the transition to more electric aircraft, one of the technological locks is the overheating of electronic components, which affects deeply their reliability and their lifetimes. Therefore, it is necessary to control the temperature of the electronic components. This thesis aims to build two reduced thermal models to monitor in real time the temperature of electronic components . The difference between the direct reduced model (DROM) and the inverse reduced order model (IROM) lies in the input parameters and the mathematical formalism of their constructions.

For the DROM, the input parameters are the boundary conditions. This model is developed in two stages. The first step is to build a reduced base representative of the solution by applying POD (Proper Orthogonal Decomposition) on a snapshots matrix. The snapshots matrix is obtained from the finite element model (FEM) solution. The second step is to calculate the coordinates of a new solution using the Galerkin projection of the FEM on the reduced basis. A DROM built with 10 modes decreases drastically the computational time and the obtained absolute error is less than $0.1\ ^{circ}C$ except during sudden power variations.

For the IROM, the input parameters are the temperature of the sensors placed far from the electronic components. This model does not need to know the boundary conditions as the DROM. The first step is to build a reduced base that couples the temperature or electronic components and the temperature of sensors using the POD. The second step is to identify the coordinates of the electronic components temperature from the measurements using a minimization algorithm. The error of the IROM of order 3 does not exceed $0.6\ ^{circ}C$ except during sudden power variations.

Keywords: Model reduction, POD (Proper Orthogonal Decomposition), heat transfer of electronic components.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : Ben Aissia DATE de SOUTENANCE : 25/09/2019

Prénom : Hazem

TITRE : Model reduction for thermal management of high power electronic components for aerospace application

NATURE : Doctorat Numéro d'ordre : 2019LYSEI071

Ecole doctorale : MEGA (Mécanique, Energétique, Génie civil, Acoustique)

Spécialité : Thermique, Energétique

RESUME :

Dans la transition vers l'avion plus électrique, un des verrous technologiques est l'échauffement des composants électroniques ce qui affecte fortement leurs fiabilités et leurs durées de vie. Par conséquence, il est nécessaire de contrôler la température des cartes électroniques. Cette thèse a pour objectif de construire deux modèles thermiques réduits permettant de suivre en temps réel la température des composants électroniques. La différence entre le modèle réduit directe (MRD) et le modèle réduit inverse (MRI) réside dans les paramètres d'entrée et le formalisme mathématique de leurs constructions.

Pour le MRD, les paramètres d'entrée sont les conditions aux limites. Ce modèle est élaboré sur deux étapes. La première étape consiste à construire une base réduite représentative de la solution en appliquant la POD (Proper Orthogonal Decomposition) sur une matrice snapshots. La matrice snapshots regroupe la solution du modèle éléments finis (MEF).
La deuxième étape consiste à calculer les coordonnées d'une nouvelle solution en utilisant la projection de Galerkin du MEF sur la base réduite. Un MRD construit avec 10 modes diminue considérablement le temps de calcul et réalise une erreur absolue inférieure à 0,1 °C en dehors des variations brusques de puissance.

Pour le MRI, les paramètres d'entrées sont les températures des capteurs implantés loin des composants électroniques.
Ce modèle n'a pas besoin de connaitre les conditions aux limites comme le MRD. La première étape consiste à construire une base réduite qui couple les températures dans les composants électroniques et dans les capteurs de températures en utilisant la POD. La deuxième étape consiste à identifier les coordonnées des températures des composants électroniques à partir des mesures en utilisant un algorithme de minimisation. L'erreur du MRI d'ordre 3 ne dépasse pas 0,6 °C en dehors des variations brusques de puissance.

MOTS-CLÉS : Réduction de modèle, POD (Proper Orthogonal Decomposition), thermique des composants électroniques.

Laboratoire (s) de recherche : CETHIL

Directeurs de thèse: Shihe Xin, Jacques Jay

Président de jury :

Composition du jury :

Bataille, Françoise	Professeur, Université de Perpignan	Présidente
Videcoq, Etienne	MCF HDR (ENSMA Poitiers)	Rapporteur
Podvin, Bérengère	Chargé de recherche HDR (LIMSI, CNRS)	Rapporteure
Rouizi, Yassine	MCF (Université d'Evry Val d'Essonne)	Examinateur
Laraqi, Najib	Professeur (Université Paris Nanterre)	Examinateur
Giroux Julien, Stéphanie	MCF HDR (Université Claude Bernard Lyon 1)	Examinatrice
Xin, Shihe	Professeur (INSA-Lyon)	Directeur de thèse
Jay, Jacques	Professeur (INSA-Lyon)	Directeur de thèse
Knikker, Ronnie	MCF (INSA-Lyon)	Invité