



HAL
open science

Mathematical modelling and simulation of ocular blood flows and their interactions.

Lorenzo Sala

► **To cite this version:**

Lorenzo Sala. Mathematical modelling and simulation of ocular blood flows and their interactions.. Numerical Analysis [math.NA]. Université de Strasbourg, IRMA, 2019. English. NNT: . tel-02284233v1

HAL Id: tel-02284233

<https://theses.hal.science/tel-02284233v1>

Submitted on 11 Sep 2019 (v1), last revised 19 Mar 2020 (v2)

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

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

Thèse

INSTITUT DE
RECHERCHE
MATHÉMATIQUE
AVANCÉE

UMR 7501

Strasbourg

Présentée pour obtenir le grade de docteur de
l'Université de Strasbourg
Spécialité MATHÉMATIQUES APPLIQUÉES

Lorenzo Sala

**Modélisation mathématique et simulation de
flux sanguins oculaires et leur interactions**

Soutenue le 27 Septembre 2019
devant la commission d'examen

Angelo Iollo	rapporteur
Kent-Andre Mardal	rapporteur
Stéphan Cotin	examinateur
Philippe Helluy	examinateur
Philippe Moireau	examinateur
Alon Harris	invité
Christophe Prud'homme	directeur de thèse
Giovanna Guidoboni	co-directeur de thèse
Marcela Szopos	co-encadrant de thèse

irma.math.unistra.fr



Université

de Strasbourg

Contents

Notations	ix
Abstract	1
I Introduction and motivation	5
1 Introduction	7
2 Medical background	11
1 Ocular anatomy and physiology	11
2 Quantities of interest and their measurements	13
3 Glaucoma	15
II The Ocular Mathematical Virtual Simulator (OMVS)	17
3 Level 0	21
1 Nonlinear resistor model	23
2 Data and implementation	25
4 Level 1	31

1	Darcy equations with integral boundary conditions (IBCs)	32
2	Porous media model for the lamina cribrosa	33
5	Level 2	37
1	Linear elasticity	38
2	Poroelasticity	40
3	Poroelastic model for the lamina cribrosa	41
4	Biomechanics of sclera, choroid, retina and cornea	42
III	Numerical methods	47
6	Geometry and mesh	51
1	Geometry	52
2	Mesh	54
3	Further comments	56
7	3D-0D operator splitting method	57
1	Introductory notation	58
2	Energy of the system	60
3	Dimensional analysis	63
4	Operator splitting technique	64
5	Two-step splitting	65
8	Hybridizable Discontinuous Galerkin (HDG) method	71
1	Introductory notation	72
2	Error analysis	74
3	Static condensation	76
4	Connectivity of the HDG method with IBC	80
5	Linear elasticity	81

IV	Contribution to Feel++ Library	83
9	HDG method	87
1	MixedPoisson	87
2	MixedElasticity	92
3	MixedPoissonElasticity	96
4	Computational performances	98
10	3D-0D operator splitting method	105
1	Functional Mock-up Interface(FMI)	105
2	FEEL++ and Functional Mock-up Unit (FMU)	106
3	The operator splitting in FEEL++	107
11	OMVS implementation	111
1	FEEL++ interface	111
2	OMVS user friendly interface	118
V	Simulation results	129
12	Verification	133
1	HDG convergence study	133
2	Operator splitting convergence study	145
13	Validation	159
1	Hemodynamics of the ocular posterior segment	160
2	Biomechanics of the lamina cribrosa	162
3	Biomechanics of cornea, sclera, choroid and retina.	164
14	Blood pressure study	169
1	Data	169
2	Results	170

3	Conclusions	172
15	Intraocular pressure study	173
1	Data	173
2	Results	174
3	Conclusions	175
16	Trans-laminar pressure difference study	177
1	Data	177
2	Results	178
3	Conclusions	180
17	Online database	183
1	Origin of data	183
2	Data of 10 virtual patients	184
3	Simulation results	184
4	Conclusions	187
VI	Sensitivity analysis	189
18	Introduction to sensitivity analysis	193
1	Sobol index sensitivity analysis	194
2	Polynomial Chaos Expansion	196
3	Discussion on the IOP input parameter	198
19	Preliminary uncertainty propagation analysis	201
1	Data	201
2	Implementation	202
3	Results	204
4	Conclusions	205

20 Sobol index sensitivity analysis	207
1 Data	207
2 Implementation	208
3 Results	212
4 Conclusions	215
VII Future perspectives	217
21 Cerebrospinal fluid flow	221
1 Reduced models for CSF dynamics	222
2 Distributed models for CSF dynamics	226
22 Ballistocardiography for cardiovascular monitoring	231
1 Background and related work	233
2 Mathematical Formulation of Closed-Loop Model	234
3 Results	241
4 Theoretical and experimental ballistocardiograms	244
5 Conclusions	246
23 nMOS transistor	247
1 Electronic model	247
2 Data	249
3 Results	251
4 Conclusions	253
24 Traction test study	255
1 Data	255
2 Results	257
3 Conclusions	261

VIII Conclusions and outlook	263
25 Conclusions and outlook	265
Bibliography	267
Presentations and publications during the period of doctoral studies	287

Notations

BC		Boundary Condition
BCG		BallistoCardioGram
BDF		Backward Differentiation Formula
BP		Blood Pressure
CAD		Computer Aided Design
CBF		Cerebral Blood Flow
CCT		Central Corneal Thickness
CDI		Color Doppler Imaging
CG		Continuous Galerkin
CNS		Central Nervous System
CO		Cardiac Output
CRA		Central Retinal Artery
CRV		Central Retinal Vein
CSF		CerebroSpinal Fluid
DAE		Differential Algebraic Equations
DOF		Degree of freedom
DP		Diastolic blood Pressure
DSEL		Domain Specific Embedded Language
ECG		ElectroCardioGram
ECS		Extracellular Space
EDV		End-Diastolic Volume
EF		Ejection Fraction
EMA		European Medicines Agency
FDA		U.S. Food & Drug Administration
ESV		End-Systolic Volume
FEM		Finite Element Method

FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GASM	Generalized Additive Schwarz Method
GAMG	Geometric Algebraic MultiGrid
HPC	High Performance Computing
HR	Heart Rate
IBC	Integral Boundary Condition
ICP	IntraCranial Pressure
IIH	Idiopathic Intracranial Hypertension
IOP	IntraOcular Pressure
JSON	JavaScript Object Notation
LC	Lamina cribrosa
MAP	Mean Arterial Pressure
(mi)CFD	(medical image-based) Computational Fluid Dynamics
MSO4SC	Mathematical Modelling, Simulation and Optimization for Societal Challenges with Scientific Computing
NDD	Neuro-Degenerative Diseases
NTG	Normal Tension Glaucoma
ODEs	Ordinary Differential Equations
OMVS	Ocular Mathematical Virtual Simulator
PC	Polynomial Chaos
(PC-)MRI	(Phase Contrast -) Magnetic Resonance Imaging
PDE	Partial Differential Equation
PDF	Probability Density Function
PVI	Pressure-Volume Index
RGCs	Retinal Gangliol Cells
RLTp	RetroLaminar Tissue pressure
SAS	SubArachnoid Space
SP	Systolic blood Pressure
SV	Stroke Volume
TLPd	TransLaminar Pressure difference
VVUQ	Verification Validation and Uncertainty Quantification
0D	Zero-Dimensional
1D	One-Dimensional
3D	Three-Dimensional

Optic neuropathies such as glaucoma are often late-onset, progressive and incurable diseases. Despite the recent progress in clinical research, there are still numerous open questions regarding the etiology of these disorders and their pathophysiology. Furthermore, data on ocular posterior tissues are difficult to estimate noninvasively and their clinical interpretation remains challenging due to the interaction among multiple factors that are not easily isolated. The recent use of mathematical models applied to biomedical problems has helped unveiling complex mechanisms of the human physiology.

In this very compelling context, our contribution is devoted to designing a mathematical and computational model coupling tissue perfusion and biomechanics within the human eye. In this thesis we have developed a patient-specific Ocular Mathematical Virtual Simulator (OMVS), which is able to disentangle multiscale and multiphysics factors in a accessible environment by employing advanced and innovative mathematical models and numerical methods. Moreover, the proposed framework may serve as a complementary method for data analysis and visualization for clinical and experimental research, and a training application for educational purposes.

In the first part of the thesis, we describe the anatomy of the eye and the pathophysiology of glaucoma. Next, we define the modeling choices and the mathematical architecture of the OMVS (Part II). In part III we present the complex ocular geometry and mesh along with the new numerical methods we have developed, namely the Hybridizable Discontinuous Galerkin method with Integral Boundary Conditions, and the operator splitting technique for solving coupled PDE-ODE systems. The fourth part of the thesis gathers all the C++ libraries that have been implemented to create and solve the OMVS. Part V illustrates the OMVS simulation results, specifically the verification and the validation strategy, and some clinically meaningful virtual experiments. Then, we propose a preliminary uncertainty quantification study, in particular an analysis on the propagation of uncertainties and a sensitivity analysis using Sobol indices (Part VI). Finally, in the last part of the thesis, we draw the conclusions and characterize different projects that can be integrated in the OMVS in the future.

Les neuropathies optiques comme le glaucome sont souvent des maladies tardives, évolutives et incurables. Malgré les progrès récents de la recherche clinique, de nombreuses questions relatives à l'étiologie de ces troubles et à leur physiopathologie restent ouvertes. De plus, les données sur les tissus postérieurs oculaires sont difficiles à estimer de façon non invasive et leur interprétation clinique demeure difficile en raison de l'interaction entre de multiples facteurs qui ne peuvent pas être facilement isolés. L'utilisation récente de modèles mathématiques pour des problèmes biomédicaux a permis de révéler des mécanismes complexes de la physiologie humaine.

Dans ce contexte très enthousiasmant, notre contribution est consacrée à la conception d'un modèle mathématique et computationnel couplant l'hémodynamique et la biomécanique de l'œil humain. Dans le cadre de cette thèse, nous avons mis au point un modèle spécifique au patient appelé simulateur virtuel de mathématiques oculaires (OMVS), capable de démêler les facteurs multi-échelles et multi-physiques dans un environnement accessible en utilisant des modèles mathématiques et des méthodes numériques avancés et innovants. De plus, le cadre proposé peut servir comme méthode complémentaire pour l'analyse et la visualisation des données pour la recherche clinique et expérimentale, et comme outil de formation pour la recherche pédagogique.

Dans la première partie de la thèse, nous décrivons l'anatomie de l'œil et la physiopathologie du glaucome. Ensuite, nous définissons les choix de modélisation et l'architecture mathématique de l'OMVS (partie II). Dans la partie III, nous présentons la complexe géométrie oculaire et le maillage computationnel ainsi que les nouvelles méthodes numériques que nous avons développées, à savoir la méthode de Galerkin Discontinue Hybride avec conditions limites intégrales, et la technique de décomposition d'opérateur pour résoudre les systèmes EDP-EDO couplés. La quatrième partie de la thèse rassemble toutes les bibliothèques C++ qui ont été implémentées pour créer et résoudre l'OMVS. La partie V illustre les résultats de la simulation de l'OMVS, en particulier la stratégie de vérification et de validation, ainsi que certaines expériences virtuelles significatives sur le plan clinique. Ensuite, nous proposons une étude préliminaire de quantification d'incertitude, notamment une analyse sur la propagation des incertitudes et une analyse de sensibilité à l'aide des indices de Sobol (partie VI). Enfin, dans la dernière partie de la thèse, nous en tirons les conclusions et caractérisons différents projets qui pourront être intégrés dans l'OMVS à l'avenir.

Part I

Introduction and motivation

CHAPTER 1

Introduction

The interest on mathematical models applied to biomedical problems has greatly increased. This situation emerges from the incredible progress that has been made in medicine in the last years, especially thanks to the advancements in the technology that has allowed for analyses that were impossible just a few years ago. For instance, (i) robotic surgery helps to aid in precision, control and flexibility with a minimal invasive procedure [156], (ii) 3D printing can be used to create implants [165] or special pills that may contain multiple drugs, which will help patients with the organization, timing and monitoring of multiple medications [36], and (iii) virtual reality had a great positive impact on the medical students training [115] and in the patient rehabilitation and recovery [54].

In this context, clinicians are continuously looking for new techniques to obtain patient-specific information in a non-invasive way, in particular exploiting mathematical models. This unceasing growth has raised exponentially the complexity of mathematics that is employed for the description of such physical systems.

Thanks to its accessibility to measurements and its special connection, the eye provides a unique window on the brain, thereby offering non-invasive access to a large set of potential biomarkers that might help in the early diagnosis and clinical care of Neuro-Degenerative Diseases (NDD). However, characterizing ocular biomarkers as surrogates of cerebral or systemic vascular status is far from trivial. In this thesis, we focus on glaucoma, which is an ocular NDD that involves the optic nerve head, by that providing a first step towards the study of the system eye-brain.

Clinical measurements are influenced by many factors that vary among individuals and cannot be isolated *in vivo*, thereby posing serious challenges for the interpretation of such measurements. This difficult, yet extremely appealing opportunity of using the eye as a window on the brain provides the main rationale of our contribution to the project. More specifically, stems from the basic ideas that an ocular measurement *per se* does not allow to draw any con-

clusion on what might be the fluid-dynamical and/or metabolic status of the brain in a given patient, unless some other factors specific to that patient are properly taken into account. In this framework mathematical modeling can provide quantitative tools to help accounting for patient-specific factors when interpreting potential ocular biomarkers.

The use of mathematical and computational models to study biofluids is not a novelty. The cardiovascular system was one of the first human systems to be studied with this kind of approach [98, 172]; later also other organs were investigated, such as lungs [135, 178] and brain [57, 153]. On the contrary, the interest on the eye is very recent, with many opportunities of developments.

Thus, motivated by the need of mathematical and computational methods to study the Eye-Brain system - which we refer to as Eye2Brain - and aid the interpretation of ocular measurements as biomarkers for the brain status, this PhD thesis relates to the development of a multi-component platform combining detailed descriptions of the eye in view of a connection with the brain network. Specifically the Eye2Brain project aims at studying the interactions among the biofluids within the eye and the brain (aqueous and vitreous humor, blood, cerebrospinal fluid), the biomechanics of the ocular tissues (retina, sclera, cornea), the electric activity and the metabolism within the neurons.

Such level of complexity calls for a multiscale and multiphysics modeling approach. Network based models allow to capture the main dynamics of complex systems at relatively low computational costs, whereas detailed 3d models allow to interface with clinical data that are 3d in nature, *e.g.* images obtained with magnetic resonance or optical coherence tomography. This thesis is a first attempt to realize this ambitious project and it focuses on the hemodynamics of the ocular posterior segment and the biomechanics of the main tissues of the eye. In order to accomplish this task, we have theorized and implemented innovative numerical methods, which handle the multiscale and multiphysics (PDEs/ODEs coupling) in a High Performance Computing (HPC) framework.

The development of this articulated computational platform that can simultaneously process and integrate medical images and measurements obtained with various instruments on the same patient bears a tremendous importance from the clinical viewpoint. Despite the significant advances in medical imaging, it is still extremely challenging for the attending physicians to have a clear full picture of the clinical status of the patient.

This is due mainly to two reasons:

- each instrument targets different parts and functions of the patient's tissue; however, the relative significance of various measurements might depend on the patient's status;
- many instruments do come with specific softwares that attempt to classify the measured data as normal, suspicious or abnormal; however, these softwares are based on different cohorts for health and disease, thereby making it difficult to combine the outcomes.

Thus, the development of an articulated platform capable of providing physicians with an integrated view of the patient's status will significantly improve their current ability to monitor health and to prevent, detect, treat and manage disease in a personalized manner.

Within this thesis, we propose to develop the mathematical model behind this complex physiological system, in particular a software for application in ophthalmology. The goal is to implement, test and deliver a software that can be used in clinics to improve diagnosis and care

of ocular diseases (*e.g.* glaucoma, diabetic retinopathy, age-related macular degeneration) and other pathologies that also manifest in the eye (*e.g.* diabetes, hypertension, NDD) [137].

Thanks to the collaboration with Prof. Harris in Indianapolis we had the opportunity to compare and test our ideas and predictions with hundreds of prospective investigations involving ocular structure, ocular perfusion and blood flow in health and disease, including diabetes, glaucoma and age-related macular degeneration conducted over the past several decades at the Eugene and Marilyn Glick Eye Institute (Indianapolis, USA).

The thesis is structured as follows:

Part I: we describe the anatomy of the eye and the pathophysiology of glaucoma (Ch. 2).

Part II: we define the modeling choices and the mathematical architecture of the OMVS, in particular the three stages of complexity of the model: *Level 0* (Ch. 3), *Level 1* (Ch. 4), and *Level 2* (Ch. 5).

Part III: we present the complex ocular geometry and mesh (Ch. 6) along with the new numerical methods we have developed, namely the operator splitting technique for solving coupled PDE-ODE systems (Ch. 7), and the Hybridizable Discontinuous Galerkin (HDG) method with Integral Boundary Conditions (Ch. 8).

Part IV: we gather all the C++ libraries that have been implemented to create and solve the OMVS, *i.e.* the HDG toolboxes (Ch. 9), the 3D-0D coupling libraries (Ch. 10), and finally the developer and user-friendly OMVS interfaces (Ch. 11). The purpose of this part is to provide a one-to-one correspondence between the math, described in Part III, the code and the results (Part V).

Part V: we illustrate the OMVS simulation results, in particular

- the verification *via* a convergence study for the HDG method and 3D-0D operator splitting technique (Ch. 12);
- the validation strategy (Ch. 13);
- a virtual study to understand the effect of the blood pressure on the ocular system (Ch. 14);
- a virtual study to understand the effect of the intraocular pressure on the ocular system (Ch. 15);
- a virtual study to understand the effect of the trans-laminar pressure difference on the ocular system (Ch. 16);
- a study on a virtual patient database using the OMVS as clinical investigative tool (Ch. 17).

Part VI: we propose a preliminary uncertainty quantification study (Ch. 18), in particular an analysis on the propagation of uncertainties (Ch. 19) and a sensitivity analysis using Sobol indices (Ch. 20).

Part VII: we characterize different projects that have been developed alongside the thesis and can be eventually integrated in the OMVS, namely:

- a review on mathematical models describing the cerebrospinal fluid flow (Ch. 21);
- a study of the cardiovascular functions in the interest of ballistocardiography (Ch. 22);

- an industrial application of the developed HDG method for Darcy equations to design a floating-gate nMOS transistor (Ch. 23);
- an application of the developed HDG method for linear elastic equations concerning material traction tests (Ch. 24).

Part VIII: we draw the conclusions together with an outlook for the future (Ch. 25).

Contents

1	Ocular anatomy and physiology	11
2	Quantities of interest and their measurements	13
3	Glaucoma	15

In this chapter we introduce the medical context of the thesis by means of three sections where we describe

- i) the physiology of the ocular tissues we have focused on;
- ii) the significant quantities of interest and their measurements;
- iii) glaucoma and the corresponding challenges in medicine.

1 Ocular anatomy and physiology

Fig. 2.1 illustrates a scheme of the ocular anatomy.

The eye is a complex organ composed by multi-layer tissues (Fig. 2.1). Moving inward inside the eye, we have the *sclera*, the *choroid*, the *retina* and the *vitreous humor*. In the front of the eye we have the *cornea*, which is the main tissue that interacts with the outside, and also the anterior and posterior chambers filled with the *aqueous humor*. The aqueous and vitreous humors are separated by an ocular structure mainly constituted of *iris*, *lens* and *ligaments*.

The *sclera*, also called the white of the eye, contains mainly collagen and some elastic fibers and covers more than 80% of the surface area of the eyeball, from the cornea to the optic nerve.

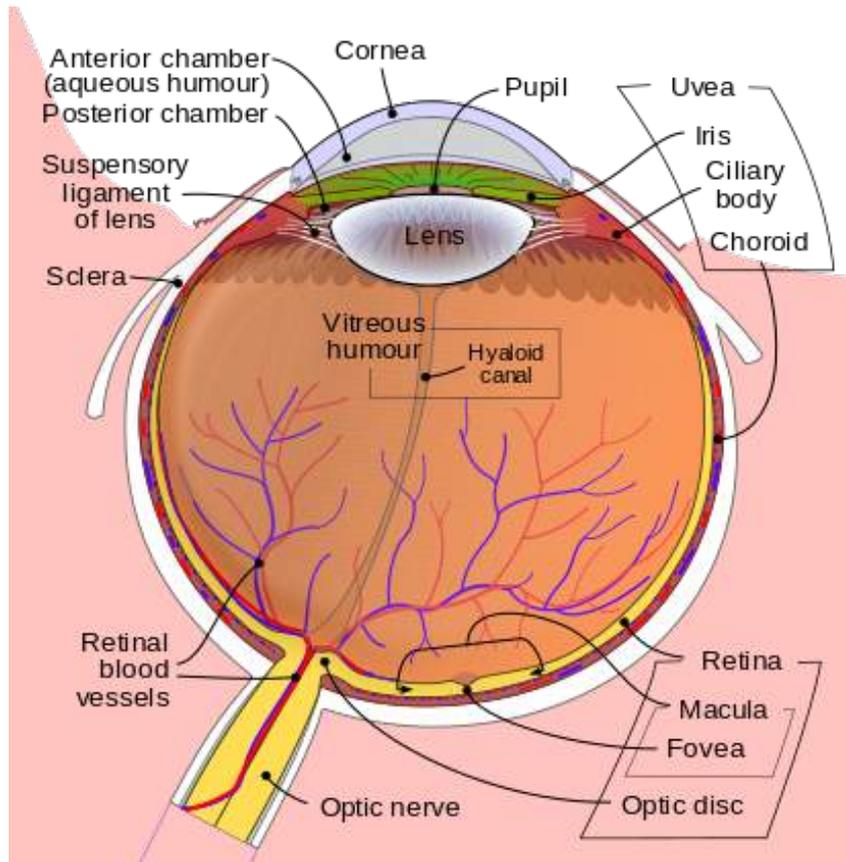


Figure 2.1 – Anatomy of the eye. Courtesy of <https://en.wikipedia.org>.

It ranges in thickness from about 0.3mm to 1.0mm and is relatively inactive metabolically with a limited blood supply. Its main function is to maintain the shape of the eye.

The *choroid* supplies the outer retina with nutrients, and maintains the temperature and volume of the eye. The choroidal circulation, which accounts for 85% of the total blood flow in the eye, is a high-flow system with relatively low oxygen content. Its main function is to conduct arteries and nerves to other structures in the eye.

The *retina* is the light sensitive inner layer and is composed by multiple sub-layers, which have different functions. The region that we are more interested in is where the retinal ganglion cells are located, indeed these neurons are responsible for the transmission of the visual information from the retina to the brain *via* the optic nerve.

The ocular blood supply comes mainly from the *ophthalmic artery* (OA), which is located close to the optic nerve. From the OA, the *central retinal artery* (CRA) branches off and enters the optic nerve approximately 10mm behind the eyeball.

In parallel with the *central retinal vein* (CRV) that drains the blood from the eye, the CRA runs within the central lumen of the optic nerve till the optic nerve head or lamina cribrosa. The CRA branches into four major arteries that nourish the four quadrant of the retina, respectively. From the retinal microvasculature, the blood is drained *via* four major retinal veins that converge into the CRV.

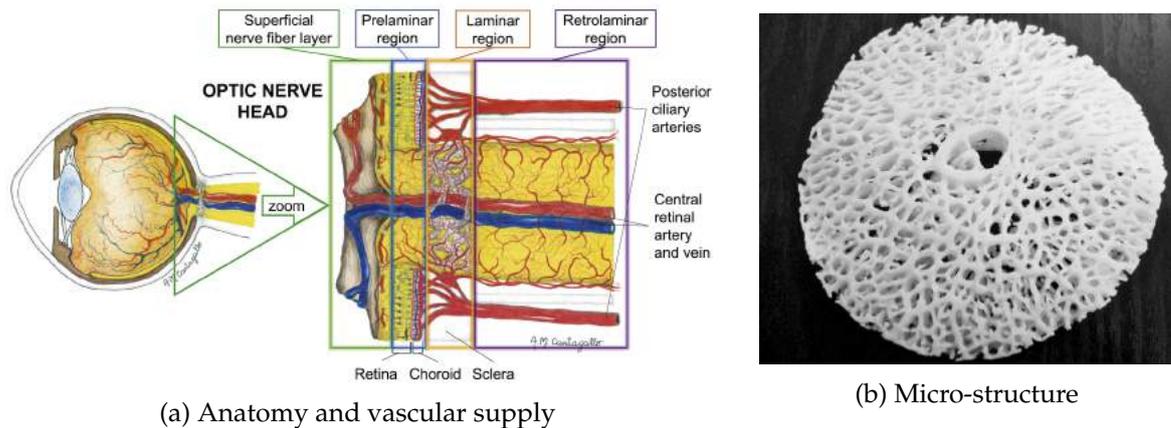


Figure 2.2 – Lamina cribrosa. Panel (a) shows the anatomy and the vascular supply. Courtesy of A.M. Cantagallo [184]. Panel (b) illustrates the micro-structure. Courtesy of <https://coggle.it>.

The *lamina cribrosa* is an extension of the sclera, which allows the retinal ganglion cells and the central retinal vessels to access the eyeball protected environment. Its anatomical position and elevated vascular supply are illustrated by Fig. 2.2a. Its biomechanical response is mainly driven by collagen beams that forms a network that gives strength to this tissue but at the same time let the nerve fibers pass through it (Fig. 2.2b). The lamina cribrosa has, therefore, a crucial role in the connection between the eye and the brain - from a neurological viewpoint - and between the eye and the cardiovascular system - from a hemodynamical viewpoint. Its micro-structure is still a matter of debate in clinical research [63, 226, 88, 245], however very recently, in some works proposed by Brazile et al. [22], the authors have discovered, thanks to highly detailed images that, on average,

- i) only 22% of the collagen beams contain a vessel within;
- ii) 21% of the vessels within the lamina cribrosa were located outside of a collagenous beam.

Besides, the authors also affirm that the distribution of blood vessels doesn't have a significant difference between the various subregions of the lamina.

2 Quantities of interest and their measurements

In this section we introduce the main quantities of interest in the eye in view of the description of the mathematical model in Part II. Moreover, we present the measurement techniques to retrieve the data for these quantities giving some insights on their accuracy and variability. The overview of these quantities and their measurements is illustrated by Fig. 2.3.

One of the most important quantity for our future investigation is the *translaminal tissue pressure difference* (TLPd), which is the pressure difference between the *intraocular pressure* (IOP) and the *retrolaminar tissue pressure* (RLTp):

$$TLPd = IOP - RLTp.$$

IOP is the pressure of the fluids inside the eye and it is determined by the balance between the production and the drainage of aqueous humor. This pressure is measured with the Goldmann Applanation Tonometer, which measures the force necessary to flatten an area of the cornea,

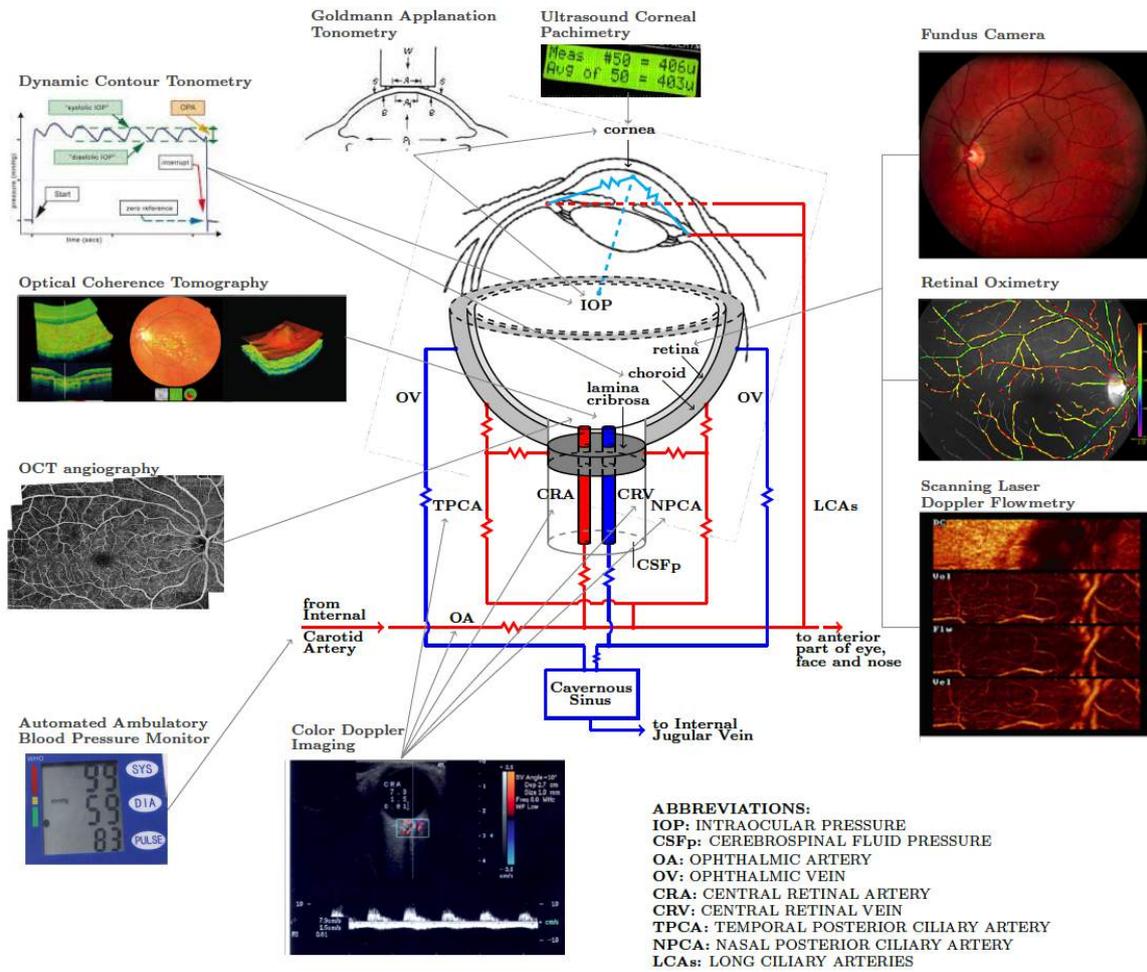


Figure 2.3 – Overview scheme of the main quantities of interest and the instruments to measure them. Courtesy of the Eugene and Marilyn Glick Eye Institute (Indianapolis, USA).

computing therefore the pressure exploiting the Imbert-Fick law [64]. This measurement accuracy is however affected by the central corneal thickness (CCT) and the corneal rigidity, in particular, thicker CCT may give artificially high IOP measurements, whereas thinner CCT may give artificially low readings [14]. The dynamics contour tonometer partially solves this issue [131], indeed it is based on the principle that when the tip of the device exactly matches the contour of the cornea, the pressure measured by a transducer placed on its tip is an accurate indicator of the true IOP [187].

The RLTP is the pressure of the tissues behind the lamina cribrosa. This pressure is largely determined by the pressure within the subarachnoid space, *i.e.* the intracranial pressure (ICP)[161]. ICP is usually measured with a lumbar puncture, an invasive technique in which a needle is inserted into the spinal canal to collect cerebrospinal fluid (CSF). Some recent works in medicine [162, 198, 159] and with mathematical models [78], however, show some innovative methods in order to estimate ICP noninvasively.

For what concerns the hemodynamics, the measurements that are relevant for the description and validation of our model are mainly the *systemic systolic* and *diastolic pressures* (SP/DP), the mean arterial pressure (MAP), and the *retinal blood flow*. The sphygmomanometer is a clin-

ical instrument, which typically consists of an inflatable rubber cuff that is applied to the arm and connected to a column of mercury next to a graduated scale, enabling the determination of systolic and diastolic blood pressure by increasing and gradually releasing the pressure in the cuff. For the retinal blood flow measurements we refer to the Color Doppler Imaging (CDI) or Doppler ultrasonography [251], which employs the Doppler effect to generate imaging of the movement of tissues and body fluids - usually blood - and their relative velocity to the probe by calculating the frequency shift of a particular sample volume. For the retinal blood flow, another quantity that the ophthalmologist can measure is the oxygenation of the blood by means of the retinal oxymetry [99].

Hence we discuss the *geometric* and *material properties* of the ocular tissues. Thanks to the Optical Coherence Tomography Angiography (OCTA), we have access to the dimensions, the thickness, and vessel structure of the lamina and the retina [148, 259]. With the pachymeter, instead, we can measure the CCT. For the material properties we point out the use of the ultrasound elastography and we refer to the abundant literature in ocular biomechanics, in particular to some *ex-vivo* measurements and traction tests with advanced mathematical analysis [111, 175, 169].

Finally, we remark that not all the clinical tools we have mentioned above are available in all the research centers or hospitals, in particular we refer to the dynamics contour tonometer, the Eco Doppler Imaging, the retinal oxymetry and the ultrasound elastography. The use of the OCTA is becoming quite ordinary for ophthalmologists working on glaucomatous patients, which is moderately replacing the OCT (Optical Coherence Tomography).

We emphasize that the presented overview does not aim at being exhaustive in the wide clinical context, rather to give some ideas of the instruments and their accuracy that has to be considered when using the corresponding data for mathematical modeling and the measurements that are not yet directly accessible nowadays.

3 Glaucoma

Glaucoma is an optic neuropathy characterized by progressive death of retinal ganglion cells (RGCs) and irreversible vision loss. The only treatable risk factor for glaucoma is elevated IOP, however, for instance, Normal Tension Glaucoma (NTG) patients show a disease progression even with IOP lower than the threshold approved to use treatments. Another interesting clinical situation is ocular hypertension (OHT). In this case, patients suffer from consistently elevated IOP in one or both eyes in the absence of clinical evidence of optic nerve damage, visual field defect or other pathology that could explain high IOP.

Thus leads to the conclusion that IOP cannot be the only factor in glaucoma. Recent works have emphasized that other confounding factors have to be considered in the study of glaucoma, for instance:

- TLPd, *i.e.* the combined interaction of IOP and RLTP [247, 150];
- ocular shape, for instance the axial length [77, 106];
- CCT, which can largely alter the IOP measurements [21].

In addition to these factors that are local to the eye, various studies suggest that also systemic factors may have a significant impact on the ocular mechanisms, such that:

- blood pressure level [248, 238];
- vascular regulation [68];
- diabetes [42];
- age [100, 239, 139, 138];
- gender [242, 180];
- ethnicity [130, 224];
- body-mass index [168];
- genetic [103].

The current challenge in ophthalmology is to disentangle causes from effects for each specific patient. In this context, the OMVS may be employed to single out the mechanisms involved in the pathophysiology of glaucoma and inspect individually the influence of these multiple factors on the overall system.

Part II

The Ocular Mathematical Virtual Simulator (OMVS)

As stated in the first part of this thesis (Part I), the lamina cribrosa plays a crucial role in the balance of pressures between the eye and the brain. Since IOP has certainly a major role in glaucoma, we are interested in modeling the complex multiphysics phenomena that take place at the level of this ocular tissue.

Specifically, in this part we illustrate the evolution of the framework of the modeling and simulation environment we developed, called Ocular Mathematical Virtual Simulator (OMVS), from its early characterization to the current status.

The development of the OMVS can be organized in different stages of increasing complexity. In the following, we present these 3 mathematical modeling frameworks:

- *Level 0* describes the hemodynamics in the ocular posterior vasculature in the back of the eye *via* a reduced model based on a nonlinear ODEs system ;
- *Level 1* models the hemodynamics in the ocular posterior vasculature *via* a multiscale approach coupling a nonlinear ODEs system with the 3D Darcy equations within the lamina cribrosa;
- *Level 2* accounts for the hemodynamics in the ocular posterior vasculature *via* a nonlinear ODEs system, within the lamina cribrosa *via* 3D Darcy equations, and the biomechanics of the optic nerve head (2a), sclera, choroid, retina and cornea (2b) by means of different systems of PDEs describing their linear elastic behavior.

Details concerning *Level 0*, *Level 1*, and *Level 2* are provided in chapters 3, 4, and 5, respectively. The contents of this part has been published in [209].

CHAPTER 3

Level 0

Contents

1	Nonlinear resistor model	23
2	Data and implementation	25

Level 0 consists in a simple representation of the blood circulation in the back of the eye, in particular within the central retinal vessels, the retina and the lamina cribrosa. The blood circulation is modeled *via* a lumped-parameter model, which exploits the electric analogy to fluid flow [73]. Within this analogy, the flow of a fluid through a hydraulic network corresponds to the flow of an electric current through an electric circuit. Thus, volume, flow rate, fluid velocity, pressure correspond to electric charge, current, current density, potential, respectively (see Tab. 3.1).

	ELECTRIC	HYDRAULIC	
	conductive wire	simple pipe	
	resistor	constricted pipe	
	capacitance	pipe compliance	
q	charge	volume	V
I	current	flow rate	Q
j	current density	velocity	v
ϕ	potential	pressure	p
$j = -\sigma \nabla \phi$	Ohm's law	Poiseuille's law	$Q = \frac{\pi r^4}{8\nu} \frac{\Delta p}{l}$

Table 3.1 – Electric analogy to fluid flow

The ODEs model illustrated in Fig. 3.1 is an extension of the circuit described in [91], *i.e.* we have added the hemodynamics description of the lamina cribrosa. In the original circuit

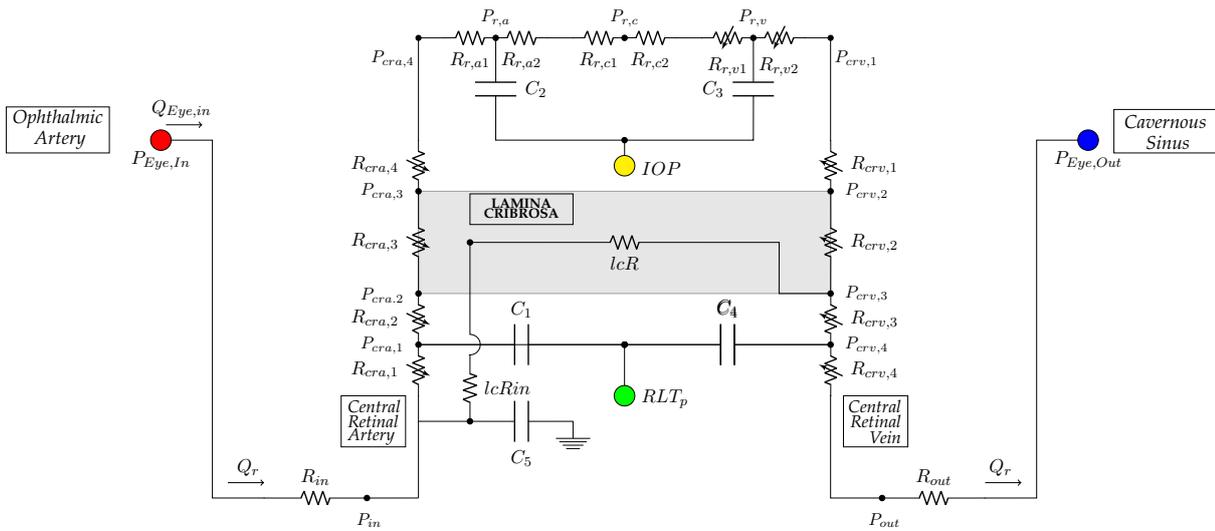


Figure 3.1 – 0D circuit representing the ocular posterior vasculature. The vasculature is divided into six main compartments: the CRA (*cra*), arterioles (*r,a*), capillaries (*r,c*), venules (*r,v*), the CRV (*crv*), and the lamina cribrosa (*lc*). Each compartment includes resistances (*R*) and capacitances (*C*). The intraocular segments ($R_{cra,3}, R_{cra,4}, R_{r,v1}, R_{r,v2}, R_{crv,1}, R_{crv,2}$) are exposed to the IOP and the retrobulbar segments are exposed to the RLTP ($R_{cra,1}, R_{cra,2}, R_{crv,3}, R_{crv,4}$).

the external pressure on the resistances $R_{cra,3}$ and $R_{crv,2}$ is the effective stress exerted by the lamina on these vessels, which has been computed *via* the simplified FSI (fluid-structure interaction) model on studied by [90] on the CRA/CRV interaction with the lamina cribrosa. In our lumped-parameter model (Fig. 3.1), we neglect this contribution by reason of the fact that we are going to introduce a more detailed 3D biomechanics description of the lamina cribrosa in *Level 2* (Ch. 5).

Following the electronic-hydraulic analogy, the 0D circuit comprises the following elements:

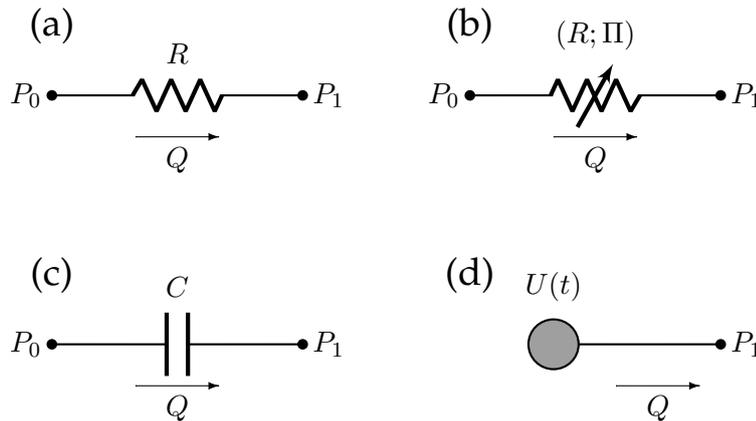


Figure 3.2 – Electrical elements included in the 0D circuit: (a) linear resistor; (b) non-linear resistor; (c) linear capacitor; (d) voltage source.

- *linear resistor* (Fig. 3.2a): the hydraulic analog to Ohm’s law states that the volumetric flow rate Q is proportional to the pressure difference $P_0 - P_1$ across the resistor, namely

$$Q = \frac{P_0 - P_1}{R} \tag{3.1}$$

where R is the hydraulic resistance;

- *nonlinear resistor* (Fig. 3.2b): this element follows the same constitutive law as the linear resistor (Eq. 3.1) with the difference that the resistance is not a constant, but it depends on a pressure Π , which is external to the element:

$$Q = \frac{P_0 - P_1}{R(\Pi)} \quad (3.2)$$

- *linear capacitor* (Fig. 3.2c): the time rate of change of the fluid volume V stored in a capacitor equals the volumetric flow rate Q . In the case of linear capacitor, the volume V is equal to the pressure difference $P_0 - P_1$ multiplied by a positive constant C called capacitance, namely $V = C(P_0 - P_1)$. Thus, the constitutive law of a linear capacitor is

$$Q = C \frac{d(P_0 - P_1)}{dt} \quad (3.3)$$

- *voltage source* (Fig. 3.2d): the hydraulic analog of a voltage source is an element that imposes the nodal pressure as $P_1(t) = U(t)$, where $U(t)$ is a given function.

1 Nonlinear resistor model

Nonlinear resistors are utilized in *Level 0* to describe the mechanical response of the CRA and the CRV to external loads. Both vessels experience two external pressures in the pre- and postlaminar regions, *i.e.* RLTp and IOP, respectively. Thus in the remaining part of this chapter the external pressure called Π in Eq. 3.2 has to be replaced by RLTp for the segments outside the globe of the eye and by IOP for the segments within the eyeball. In particular, our intent is to model the mechanical response of the vessels due to changes in the value of the transmural pressure. The transmural pressure is the difference between the pressure inside the vessel and the external load acting on the same segment.

Such analysis has already been performed in literature, and we report for sake of completeness the principal results below. For more details see [31].

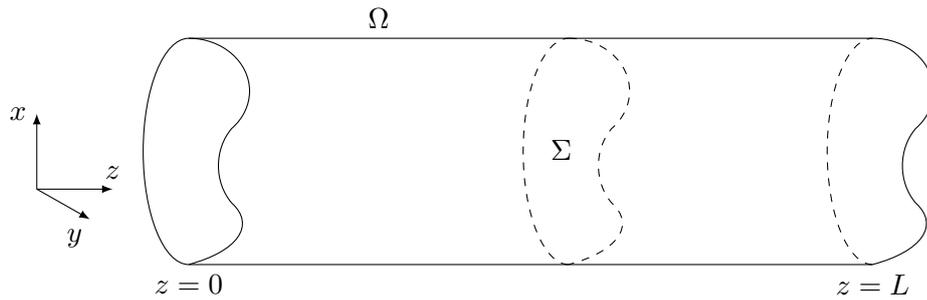


Figure 3.3 – Representative cylinder in the Cartesian coordinate system $(x; y; z)$ with cross-section Σ , of length L

Let us consider a straight cylinder Ω with cross section Σ of general shape (Fig. 3.3). Let us assume that the movement of a viscous fluid inside the cylinder can be described by the stationary Navier-Stokes equations with no-slip boundary conditions:

$$\begin{cases} \nabla \cdot \underline{u} = 0 & \text{in } \Omega \\ (\underline{u} \cdot \nabla) \underline{u} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \Delta \underline{u} & \text{in } \Omega \\ \underline{u} = \underline{0} & \text{on } \partial \Sigma \times (0, L) \end{cases} \quad (3.4)$$

where \underline{u} is the fluid velocity, p is the fluid pressure, μ is the fluid dynamic viscosity, ρ is the fluid density and $\partial\Sigma$ is the lateral boundary of the cylinder Ω . Moreover, we impose a determined pressure at inlet and outlet, and we rewrite $\underline{u} = \underline{u}_\Sigma + u_z \underline{e}_z$ with $\underline{u}_\Sigma = u_x \underline{e}_x + u_y \underline{e}_y$ and \underline{n}_Σ is the outward normal vector to $\partial\Sigma \times (0, L)$.

Let us perform a model reduction based on the following assumptions:

- p is constant on each section Σ , which means that $p = p(z)$;
- body forces and mass sources are absent;
- $|u_\Sigma| \ll |u_z|$, which means axial motion is predominant;
- $u_z = \bar{u}(z)f(\Sigma)$, where $\bar{u}(z)$ represents the average axial velocity on the cross-section Σ and $f(\Sigma)$ is an appropriate shape function;
- low Reynolds number, for instance in the CRA $Re = \frac{\rho \bar{u} L}{\mu} \simeq 4$ [31];
- the variability of A with respect to z can be neglected $\Rightarrow \int_\Sigma f(\Sigma) d\gamma = A(z)$.

Combining these assumptions we can obtain the following reduced system:

$$\begin{cases} \frac{dQ(z)}{dz} = 0 & \text{in } (0, L) \\ \frac{1}{K_r \rho} \frac{dp}{dz} + \frac{Q(z)}{A^2(z)} = 0 & \text{in } (0, L) \end{cases} \quad (3.5)$$

where $Q(z) = \int_\Sigma u_z d\sigma = \bar{u}(z)A(z)$ is the volumetric average flow rate, $A(z) = \int_\Sigma 1 d\sigma$ is the cross-sectional area and $K_r = -\frac{\mu}{\rho} \int_{\partial\Sigma} \underline{n}_\Sigma \cdot \nabla_\Sigma f(\Sigma) d\gamma$. The choice of the specific shape function $f(\Sigma)$ for the fluid motion influences the value of K_r . For Poiseuille's flow, where the fluid velocity has a parabolic profile, we have that $K_r = \frac{8\pi\mu}{\rho}$ [190, 27].

Till now, we have not introduced yet any mechanical interaction between the vessels and the fluid, in particular depending on the transmural pressure difference. Thus, we adopt the tube law [27, 86, 181, 219]:

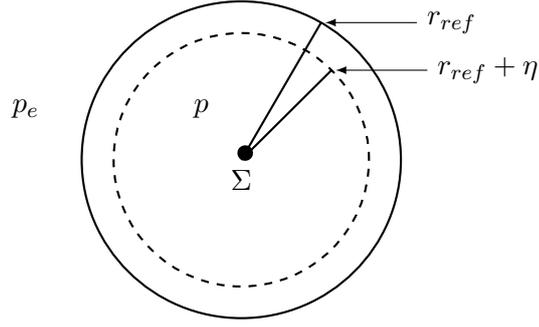
$$P(\alpha) = \frac{p(z) - p_e}{K_p} \quad (3.6)$$

where $\alpha = \frac{A(z)}{A_{ref}}$ is the ratio between the cross-sectional area $A(z)$ and the reference cross-sectional area A_{ref} . $p(z)$ and p_e are the internal and external pressure acting on the tube, respectively, and K_p is a material parameter that, for a linear elastic tube, can be expressed as

$$K_p = \frac{E}{12(1 - \nu^2) \left(\frac{h}{r_{ref}} \right)^3}$$

where E , ν , h and r_{ref} are the tube Young modulus, Poisson's ration, wall thickness and reference radius, respectively [86, 181, 219].

Thus, we have two possible situations, *i.e.* cross-section constriction ($p_e > p(z) \Rightarrow \alpha < 1$, Fig.

Figure 3.4 – Radial displacement on cross-section Σ for compressible tube in the case $\alpha < 0$

3.4) and cross-section dilation ($p_e < p(z) \Rightarrow \alpha > 1$). The situation $\alpha = 1$ corresponds to circular cross-section: $A(z) = A_{ref} = \pi r_{ref}^2$.

For the sake of simplicity hereafter we will consider the retinal venules as part of the CRV segment. Following the passages described in [31] and [207] we adopt the following model choices:

$$\text{CRA} \quad R = \frac{K_r \rho L}{A_{ref}^2} \left[\frac{\hat{p} - p_e}{K_p k_L} + 1 \right]^{-4} \quad (3.7)$$

$$\text{CRV} \quad R = \begin{cases} \frac{K_r \rho L}{A_{ref}^2} \left[\frac{\hat{p} - p_e}{K_p k_L} + 1 \right]^{-4} & \hat{p} \geq p_e \\ \frac{K_r \rho L}{A_{ref}^2} \left[1 - \frac{\hat{p} - p_e}{K_p} \right]^{4/3} & \hat{p} < p_e \end{cases} \quad (3.8)$$

where L is the length of the vessel, \hat{p} is the average pressure inside the vessel and $k_L = 12 \left(\frac{r_{ref}}{h} \right)^2$.

In Eq. 3.7 we have assumed that the cross-section remains circular in presence of external pressures, whereas in Eq. (3.8) we have employed the *Starling resistor effect* based on experiments suggested by [27, 219]. This formula expresses that, in case of compressible tubes, the circular cross-section is preserved for dilation, whereas when the transmural pressure is negative the cross-section becomes elliptical.

In conclusion, we have defined two types of nonlinear resistors - for CRA and CRV segments, respectively - that are univocally characterized by three parameters: $k_0 = \frac{A_{ref}^2}{K_r \rho L}$, k_L and K_p .

2 Data and implementation

In Tabs. 3.2 and 3.3 we report the parameter values we have used in the simulation of this first model, if not specified differently in the results (Part V). The data are taken from [31].

The ODEs model has been implemented in OpenModelica [76], Fig. 3.5 shows the implementation of the model *Level 0* in this open-source software.

Name	Type	Parameters	Values	Units
P_{in}	blood pressure source	SP	120	mmHg
		DP	80	mmHg
		HR	60	mmHg
P_{out}	pressure source	V	0	mmHg
R_{LTp}	pressure source	V	7	mmHg
IOP	pressure source	V	15	mmHg
R_{in}	linear resistor	R	22500	mmHg s cm ⁻³
R_{1a}	nonlinear resistor (3.7)	k_0	$2.124 \cdot 10^{-4}$	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	55.714	[-]
		K_p	24.665	mmHg
		p_e	R_{LTp}	mmHg
R_{1b}	nonlinear resistor (3.7)	k_0	$2.107 \cdot 10^{-4}$	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	55.487	[-]
		K_p	24.816	mmHg
		p_e	R_{LTp}	mmHg
R_{1c}	nonlinear resistor (3.7)	k_0	0.0047	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	56.1468	[-]
		K_p	24.3797	mmHg
		p_e	IOP	mmHg
R_{1d}	nonlinear resistor (3.7)	k_0	0.0010	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	56.1785	[-]
		K_p	24.3591	mmHg
		p_e	IOP	mmHg
R_{2a}	linear resistor	R	6000	mmHg s cm ⁻³
R_{2b}	linear resistor	R	6000	mmHg s cm ⁻³
R_{3a}	linear resistor	R	5680	mmHg s cm ⁻³
R_{3b}	linear resistor	R	5680	mmHg s cm ⁻³
R_{4a}	nonlinear resistor (3.8)	k_0	$2.199 \cdot 10^{-4}$	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	992.4853	[-]
		K_p	0.0722	mmHg
		p_e	IOP	mmHg
R_{4b}	nonlinear resistor (3.8)	k_0	$2.199 \cdot 10^{-4}$	cm ³ mmHg ⁻¹ s ⁻¹
		k_L	992.4853	[-]
		K_p	0.0722	mmHg
		p_e	IOP	mmHg

Table 3.2 – Data for *Level 0*

Level 0 framework exploits some electronic elements already implemented in OpenModelica, in particular linear resistor, linear capacitor and pressure (voltage) source. However, in order to be able to reproduce the physiological mechanisms related to vessel collapsibility, we have implemented these new OpenModelica classes:

- **nonlinear resistor** that follows Eq. (3.8):

Name	Type	Parameters	Values	Units
R_{5a}	nonlinear resistor (3.8)	k_0	0.0031	$\text{cm}^3 \text{mmHg}^{-1} \text{s}^{-1}$
		k_L	1457.5	[-]
		K_p	0.3687	mmHg
		p_e	IOP	mmHg
R_{5b}	nonlinear resistor (3.8)	k_0	0.0156	$\text{cm}^3 \text{mmHg}^{-1} \text{s}^{-1}$
		k_L	1458.2	[-]
		K_p	0.3684	mmHg
		p_e	IOP	mmHg
R_{5c}	nonlinear resistor (3.8)	k_0	0.0007	$\text{cm}^3 \text{mmHg}^{-1} \text{s}^{-1}$
		k_L	1419.4	[-]
		K_p	0.3836	mmHg
		p_e	$RLTp$	mmHg
R_{5d}	nonlinear resistor (3.8)	k_0	0.0007	$\text{cm}^3 \text{mmHg}^{-1} \text{s}^{-1}$
		k_L	1424.1	[-]
		K_p	0.3817	mmHg
		p_e	$RLTp$	mmHg
R_{out}	linear resistor	R	14111.39	mmHg s cm^{-3}
lcR_{in}	linear resistor	R	78181.9	mmHg s cm^{-3}
$R_{bufferIN}$	linear resistor	R	2000	mmHg s cm^{-3}
lcR	linear resistor	R	21988.25	mmHg s cm^{-3}
C_1	linear capacitor	C	0.000000722	$\text{cm}^3 \text{mmHg}^{-1}$
C_2	linear capacitor	C	0.000000753	$\text{cm}^3 \text{mmHg}^{-1}$
C_3	linear capacitor	C	0.0000167	$\text{cm}^3 \text{mmHg}^{-1}$
C_4	linear capacitor	C	0.0000107	$\text{cm}^3 \text{mmHg}^{-1}$
C_5	linear capacitor	C	0.000000753	$\text{cm}^3 \text{mmHg}^{-1}$

Table 3.3 – Data for *Level 0*

CRV resistor

```

1 extends Modelica.Electrical.Analog.Interfaces.OnePort;
2 parameter Real k0(quantity = "k0", unit = "ml/(mmHg.s)") "Aref^2/(kr*ro*L), kr=8*pi*mu/ro";
3 parameter Real kl(quantity = "kl", unit = "1") "12 Aref^2/(pi*h^2)";
4 parameter Real kp(quantity = "kp", unit = "mmHg") "E/(12*(1-ni^2)*(h/rref)^3";
5 Modelica.Blocks.Interfaces.RealInput REF "Pressure that controlled the resistance";
6
7 final Modelica.SIunits.Resistance Ractual;
8 final Modelica.SIunits.Voltage DP;
9
10 equation
11   DP = n.v;
12   Ractual = noEvent( if noEvent(DP < REF)
13     then 1 / k0 * (1 - (DP - REF) / kp) ^ (4 / 3)
14     else 1 / k0 * (1 + (DP - REF) / (kp * kl)) ^ (-4) );
15
16   v = Ractual * i;

```

- **nonlinear resistor** that follows Eq. (3.7):

CRA resistor

```

1 extends Modelica.Electrical.Analog.Interfaces.OnePort;
2 parameter Real k0(quantity = "k0", unit = "ml/(mmHg.s)") "Aref^2/(kr*ro*L), kr=8*pi*mu/ro";
3 parameter Real kl(quantity = "kl", unit = "1") "12 Aref^2/(pi*h^2)";
4 parameter Real kp(quantity = "kp", unit = "mmHg") "E/(12*(1-ni^2)*(h/rref)^3";
5 Modelica.Blocks.Interfaces.RealInput REF "Pressure that controlled the resistance";
6

```

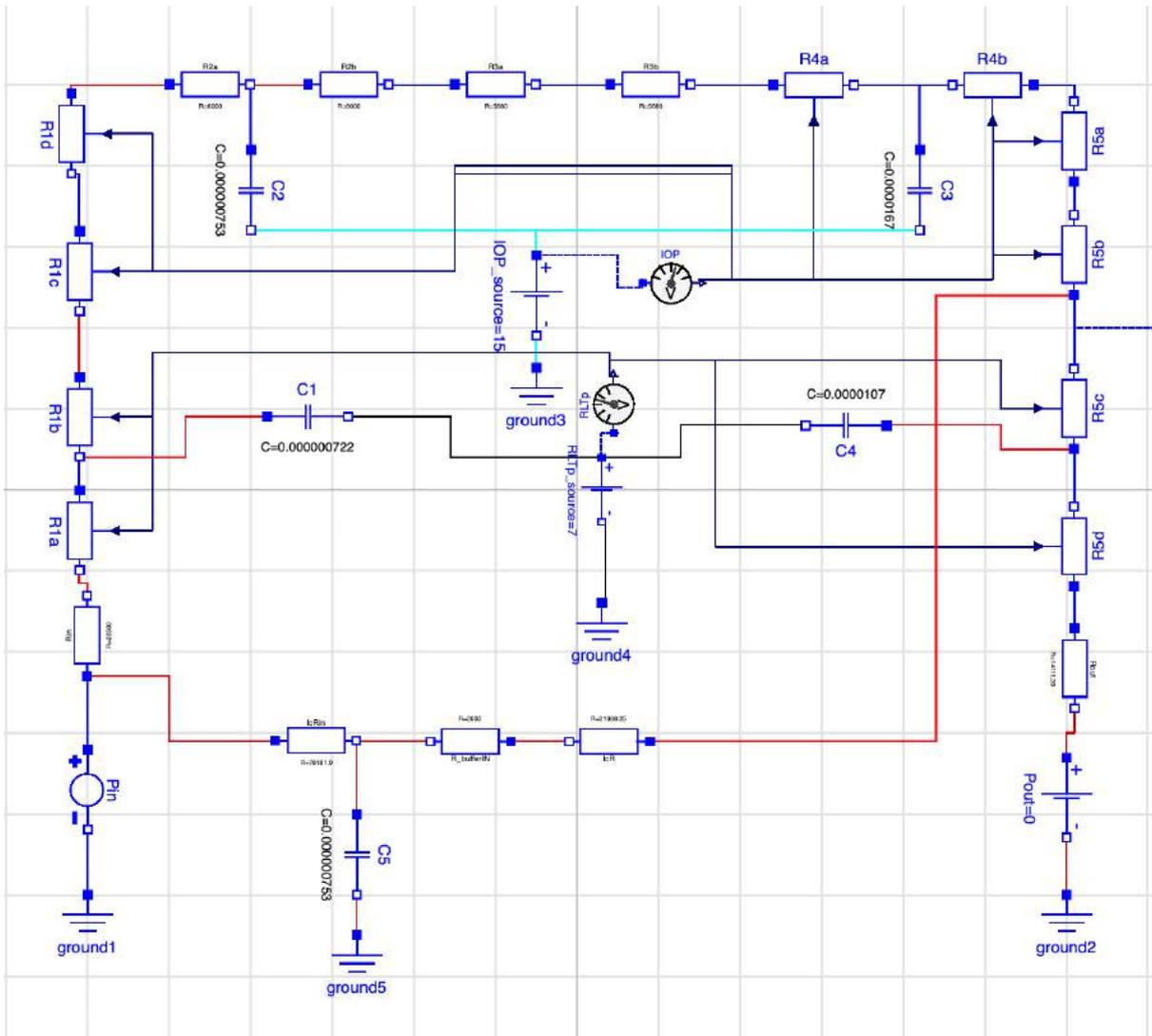


Figure 3.5 – Level 0 circuit implemented in OpenModelica

```

7   final Modelica.SIunits.Resistance Ractual;
8
9   equation
10  Ractual = 1 / k0 * (1 + ((p.v + n.v) / 2 - REF) / (kp * kl)) ^ (-4);
11  v = Ractual * i;

```

- **blood pressure source;** in this case we have employed the same approach used in [91]; we reconstruct the CRA blood pressure time profile from typical CDI measurement of the blood velocity in the CRA (Fig. 3.6) and we impose it as time-dependent pressure source. Moreover, we have parametrized the reconstructed signal using the heart rate (HR), the systemic systolic and diastolic pressures (SP and DP, respectively) as input; thus, we have

divided the CRA blood pressure time profile into 6 parts for each cardiac cycle :

$$P_{in}(t) = \begin{cases} 0.65SP - 0.475DP \sin\left(\frac{2\pi}{4 \cdot 0.082 \frac{60}{HR}} \hat{t} + \frac{2\pi}{0.328 \frac{60}{HR}} 0.082 \frac{60}{HR}\right) & \text{for } \hat{t} \leq 0.082 \frac{60}{HR} \\ 0.65SP + 0.9 \sin\left(\frac{2\pi}{0.03 \frac{60}{HR}} \hat{t} - \frac{2\pi}{0.03 \frac{60}{HR}} 0.082 \frac{60}{HR}\right) & \text{for } 0.082 \frac{60}{HR} < \hat{t} \leq 0.112 \frac{60}{HR} \\ 0.65SP + 0.118SP \sin\left(\frac{2\pi}{0.572 \frac{60}{HR}} \hat{t} - \frac{2\pi}{0.572 \frac{60}{HR}} 0.112 \frac{60}{HR}\right) & \text{for } 0.112 \frac{60}{HR} < \hat{t} \leq 0.398 \frac{60}{HR} \\ -\frac{0.13SP}{0.034 \frac{60}{HR}} \hat{t} + 0.65SP - \frac{0.13SP}{0.034 \frac{60}{HR}} 0.398 \frac{60}{HR} & \text{for } 0.398 \frac{60}{HR} < \hat{t} \leq 0.432 \frac{60}{HR} \\ 0.52SP - 0.8 \sin\left(\frac{2\pi}{0.05 \frac{60}{HR}} \hat{t} - \frac{2\pi}{0.05 \frac{60}{HR}} 0.332 \frac{60}{HR}\right) & \text{for } 0.432 \frac{60}{HR} < \hat{t} \leq 0.482 \frac{60}{HR} \\ 0.52SP + (0.52SP - 0.5DP) \sin\left(\frac{2\pi}{2.072 \frac{60}{HR}} \hat{t} + \frac{2\pi}{2.072 \frac{60}{HR}} 0.554 \frac{60}{HR}\right) & \text{for } \hat{t} > 0.482 \frac{60}{HR} \end{cases} \quad (3.9)$$

where $\hat{t} = \text{mod}\left(t, \frac{60}{HR}\right)$

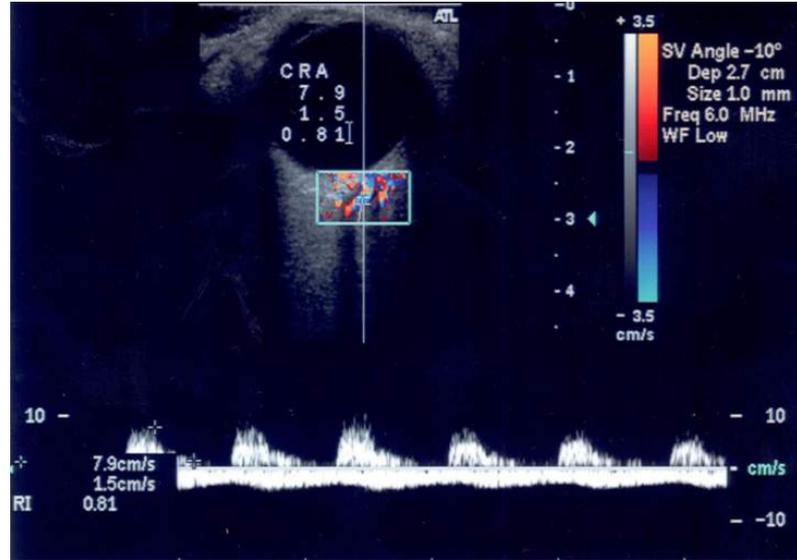


Figure 3.6 – Color Doppler Imaging (CDI) blood velocity profile within the CRA. Courtesy of the Eugene and Marilyn Glick Eye Institute (Indianapolis, USA).

Contents

1	Darcy equations with integral boundary conditions (IBCs)	32
2	Porous media model for the lamina cribrosa	33

In *Level 1* of the OMVS we look for a more detailed description of the hemodynamics within the lamina cribrosa. Since this tissue has a structure similar to a sponge - see Ch. 2 for the anatomical details of the lamina cribrosa - we have employed the Darcy equations that describe the behavior of a porous medium.

The modeling of the lamina cribrosa is not a novelty in the scientific panorama, indeed this tissue has risen a lot of interest from a biomechanical viewpoint [63, 225, 88, 167, 11, 83]. On the other hand, few studies have been completed on the hemodynamics of the lamina [40].

The proposed innovative model provides a unique window on both the biomechanics and hemodynamics. The interaction between this two aspects is investigated in Ch. 5, whereas here we focus on the importance of having an overview of the blood circulation. The originality we bring out with this model is that we consider not only the major region of interest - the lamina cribrosa - but we take into account also the blood framework from which this tissue is nourished. In this way we look at the local to systemic feedback acting on the whole system. For instance, if the lamina is compressed in a certain region, the blood flow may redistribute not only in the lamina itself, but also in the other vessels connected with this structure.

In order to accomplish this challenging task, we couple Darcy equations with the ODEs system described in Ch. 3 by using an integral boundary conditions (IBC). In the upcoming section we present the Darcy equations in a framework, which is meaningful from the viewpoint of the target application involving IBCs. We next specify the model in the medical context we are interested in. The new numerical methods that we have developed to discretize the considered

formulation are presented in Chs. 7 and 8.

1 Darcy equations with integral boundary conditions (IBCs)

The novelty of our work is the development of non-standard boundary conditions for the Darcy model written in the mixed form by reason of a natural PDEs/ODEs coupling.

Before entering the details of our application that are discussed in Sec. 2, we discuss below this mixed form formulation for Darcy from a general perspective and at a continuous level. Thus, let us denote by Ω an open bounded set of \mathbb{R}^d , $d = 2, 3$ representing the computational domain. Let the boundary Γ of Ω be partitioned into three disjoint subsets: Γ_D , Γ_N , and Γ_I . The goal is to solve the following mixed problem: find \underline{j} , p such that

$$\underline{j} + \underline{\mathcal{K}} \cdot \nabla p = 0 \quad \text{in } \Omega \quad (4.1a)$$

$$\nabla \cdot \underline{j} = f \quad \text{in } \Omega \quad (4.1b)$$

subject to the following non-standard boundary conditions

$$p = g_D \quad \text{on } \Gamma_D \quad (4.2)$$

$$\underline{j} \cdot \underline{n} = g_N \quad \text{on } \Gamma_N \quad (4.3)$$

$$\int_{\Gamma_I} \underline{j} \cdot \underline{n} = I_{target} \quad \text{on } \Gamma_I \quad (4.4)$$

$$p(\underline{x}, t) = p(t) \quad \text{on } \Gamma_I. \quad (4.5)$$

where I_{target} is a given constant. p on Γ_I is a constant (Eq. (4.5)), however its value is not known until the system is solved [13]. Moreover, we have that $f \in L^2(\Omega)$ and $\mathcal{K} \in (L^\infty(\Omega))^{n \times n}$ is a symmetric $n \times n$ matrix, uniformly positive defined over Ω and represents the permeability tensor. We use a standard notation for integrals: for functions $p, q \in L^2(D)$, we write $(p, q)_D = \int_D pq$ if $D \subset \mathbb{R}^d$, and $\langle p, q \rangle_D = \int_D pq$ if $D \subset \mathbb{R}^{d-1}$.

Introducing a fixed function $\bar{\varphi} \in H^{1/2}(\Gamma)$ verifying

$$\bar{\varphi}|_{\Gamma_D} = 0, \quad \bar{\varphi}|_{\Gamma_I} = 1,$$

we consider the following variational formulation of the problem: find $\underline{j} \in H(\text{div}; \Omega)$, $p \in L^2(\Omega)$, $\hat{p} \in \text{span}\langle \bar{\varphi} \rangle \oplus H_{00}^{1/2}(\Gamma_N)$, where

$$H_{00}^{1/2}(\Gamma_N) = \{\varphi \in H^{1/2}(\Gamma) : \varphi = 0 \text{ on } \Gamma_D \cup \Gamma_I\},$$

such that for all $\underline{v} \in H(\text{div}; \Omega)$, $w \in L^2(\Omega)$, $\mu \in \text{span}\langle \bar{\varphi} \rangle \oplus H_{00}^{1/2}(\Gamma_N)$ we have

$$(\underline{\mathcal{K}}^{-1} \underline{j}, \underline{v})_\Omega - (p, \nabla \cdot \underline{v})_\Omega + \langle \hat{p}, \underline{v} \cdot \underline{n} \rangle_\Gamma = 0, \quad (4.6a)$$

$$(\nabla \cdot \underline{j}, w)_\Omega = (f, w)_\Omega, \quad (4.6b)$$

$$\langle \underline{j} \cdot \underline{n}, \mu \rangle_{\Gamma_N \cup \Gamma_I} = \langle g_N, \mu \rangle_{\Gamma_N} + I_{target} |\Gamma_I|^{-1} \langle \mu, 1 \rangle_{\Gamma_I}, \quad (4.6c)$$

$$\langle \hat{p}, \mu \rangle_{\Gamma_D} = \langle g_D, \mu \rangle_{\Gamma_D}. \quad (4.6d)$$

This formulation leads to the following theorem:

Theorem 1. For $f \in L^2(\Omega)$, $g_D \in H^{1/2}(\Gamma_D)$, $g_N \in L^2(\Gamma_N)$, problem (4.6) admits a unique solution (j, p, \hat{p}) satisfying

$$-\nabla \cdot (\underline{\underline{K}} \nabla p) = f \quad \text{in } \Omega, \quad (4.7a)$$

$$p = g_D \quad \text{on } \Gamma_D, \quad (4.7b)$$

$$-\underline{\underline{K}} \nabla p \cdot \underline{n} = g_N \quad \text{on } \Gamma_N, \quad (4.7c)$$

$$\text{with } p \text{ constant on } \Gamma_I. \quad (4.8)$$

Moreover, the integral boundary condition

$$\int_{\Gamma_I} -\underline{\underline{K}} \nabla p \cdot \underline{n} = I_{target} \quad (4.9)$$

is satisfied.

The details and the proof of this theorem can be found in [13].

2 Porous media model for the lamina cribrosa

As anticipated in the introduction of this chapter, to describe the lamina cribrosa model, we couple the system of PDEs described above (Eq. 4.1) with the system of ODEs proposed for *Level 0* in a natural way thanks to the mixed formulation with IBCs. Specifically, we have replaced the reduced model in the gray area of Fig. 3.1 with a 3D model of the lamina. The overview of *Level 1* is sketched in Fig. 4.1.

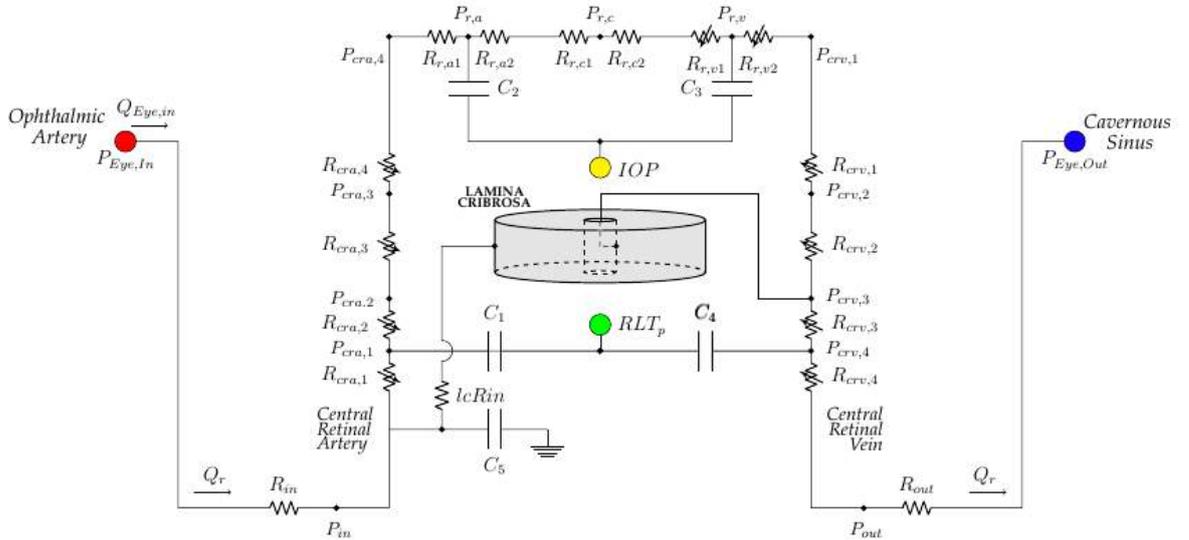


Figure 4.1 – Schematic overview of *Level 1*.

The data used for the lumped-parameter model are the same as those reported in Ch. 3, in particular we refer to Tabs. 3.2 and 3.3. For what concerns the 3D model, Fig. 4.2 depicts the boundary/interface conditions, which are an reformulation of the BCs proposed by [34]; *i.e.* we have:

- $p = p_{crv}$ on Γ_{int} ;
- $\underline{j} \cdot \underline{n} = 0$ on $\Gamma_{bottom} \cup \Gamma_{top}$, corresponding to the physiological situation where there are no outward fluxes from the top or the bottom surfaces of the lamina;
- $\int_{\Gamma_{ext}} \underline{j} \cdot \underline{n} = Q_I$ and $p|_{\Gamma_{ext}} = P_I$ on the external lateral boundary, where Q_I and P_I refer to the variables within the circuit, therefore they are not predetermined.

The BC on Γ_{ext} is crucial for the temporal and spatial coupling between *Level 0* and the new porous media model introduced in *Level 1*. This framework provides a zoom in a region of interest for the lamina cribrosa, while accounting for systemic feedback *via* the 0D circuit. In particular, the value P_I on Γ_I is obtained after solving the integral condition, which is the core to achieve the natural spatial coupling between the 3D flux (\underline{j}) and its 0D representation in the circuit (Q_I).

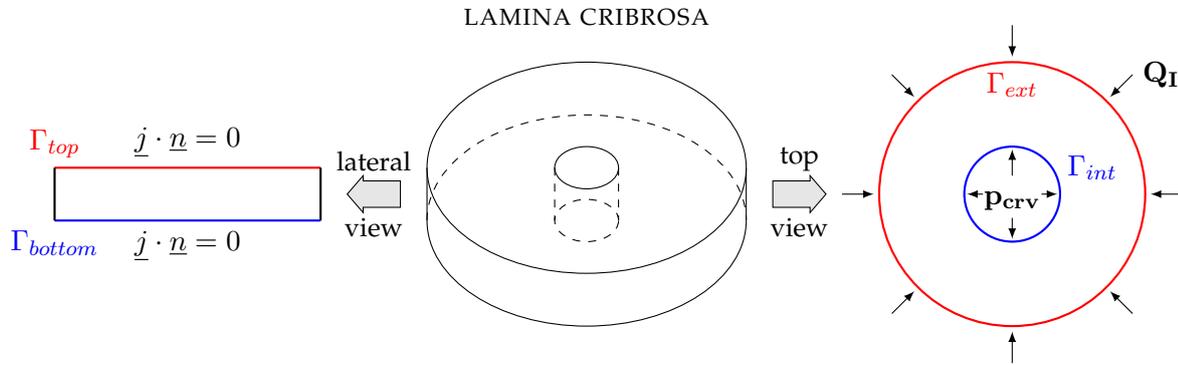


Figure 4.2 – Boundary and interface conditions for the 3D Darcy model of the lamina cribrosa.

These conditions replicate the physiology of the lamina. This tissue is nourished through the lateral boundaries, and finally the blood is gathered by the CRV at the central opening (Fig. 4.3a). We use a baseline value of $p_{crv} = 19$ mmHg [91]. We adopt the isotropic hypothesis (Fig. 4.3b) for the permeability tensor $\underline{\underline{K}}$ which takes into account for capillaries within the tissue of the lamina. Thus, $\underline{\underline{K}} = \kappa \underline{\underline{I}}$ with $\kappa = 0.015192 \text{ cm}^3 \text{ s Kg}^{-1}$. As already stated in Ch. 2, this hypothesis provides a satisfactory approximation of physiological conditions [22].

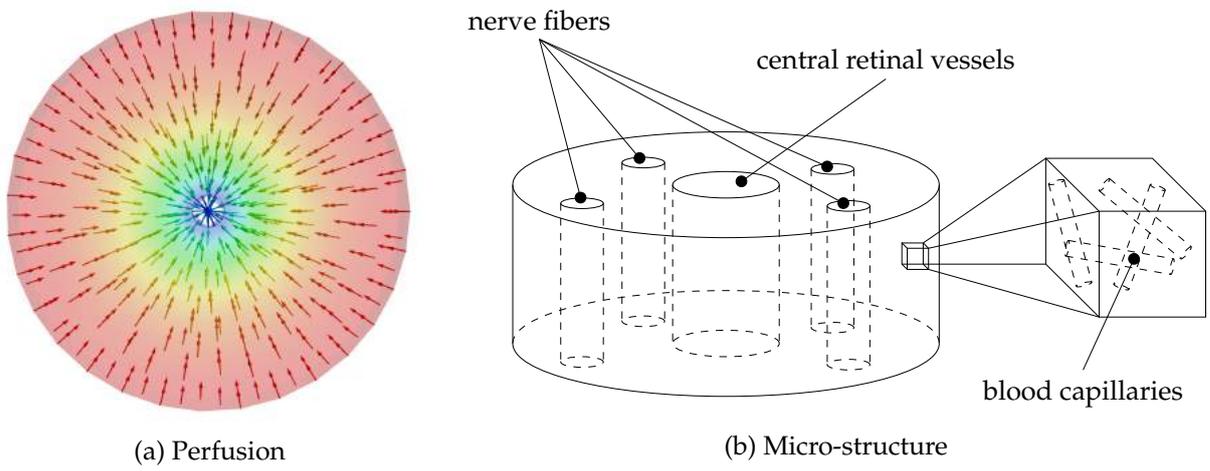


Figure 4.3 – Lamina cribrosa porous media model.

Contents

1	Linear elasticity	38
2	Poroelasticity	40
3	Poroelastic model for the lamina cribrosa	41
4	Biomechanics of sclera, choroid, retina and cornea	42

The *Level 2* of the OMVS adds to the hemodynamics of *Level 1* a biomechanical description of the lamina cribrosa, then the biomechanical part of the model is extended in a second step to the sclera, choroid, retina and cornea. The simpler version of this model is called *Level 2a*, which includes the biomechanics of the lamina cribrosa, whereas the framework that incorporates also the description of other ocular tissues is called *Level 2b*.

This chapter is structured as follows:

- a first section introduces the equations used for the biomechanical description, namely the linear elasticity system;
- a second section presents the general framework of a poroelastic model;
- a third section illustrates the specific poroelastic model of the lamina cribrosa we have used for the OMVS;
- a final section adds the description of the sclera, choroid, retina and cornea to the poroelastic system proposed above.

1 Linear elasticity

Following a similar procedure to the one utilized for the Darcy system, we present the development of integral boundary conditions for the linear elastic model written in the mixed form. Using the same notation as in Ch. 4, we consider the computational domain $\Omega \in \mathbb{R}^d$ with $d = 2, 3$, the boundary $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_I$. The scope is to solve the following mixed problem: find $\underline{\underline{\sigma}}, \underline{u}$ such that

$$\mathcal{A}\underline{\underline{\sigma}} - \underline{\underline{\varepsilon}}(\underline{u}) = \underline{0} \quad \text{in } \Omega \times [0, T] \subset \mathbb{R}^d \times \mathbb{R}, \quad (5.1a)$$

$$\rho \frac{\partial^2 \underline{u}}{\partial t^2} = \nabla \cdot \underline{\underline{\sigma}} + \underline{F}_{ext} \quad \text{in } \Omega \times [0, T], \quad (5.1b)$$

accompanied by the following boundary conditions

$$\underline{\underline{\sigma}} \cdot \underline{n} = \underline{g}_N \quad \text{on } \Gamma_N, \quad (5.2a)$$

$$\underline{u} = \underline{g}_D \quad \text{on } \Gamma_D, \quad (5.2b)$$

$$\int_{\Gamma_I} \underline{\underline{\sigma}} \cdot \underline{n} \, d\Gamma_I = \underline{F}_{target} \quad \text{on } \Gamma_I, \quad (5.2c)$$

$$\underline{u}(\underline{x}, t) = \underline{u}(t) \quad \text{on } \Gamma_I \quad (5.2d)$$

where \underline{u} is the displacement, \underline{F}_{ext} is the volumetric force term, \underline{F}_{target} is a given constant vector in \mathbb{R}^d , $\underline{\underline{\varepsilon}}(\underline{u}) = \frac{1}{2} (\nabla \underline{u} + (\nabla \underline{u})^T)$ is the strain tensor, \underline{I} is the identity tensor, $\underline{\underline{\sigma}}$ is the stress tensor - symmetric tensor in $\mathbb{R}^{d \times d}$ - and the compliance operator \mathcal{A} is assumed to be a bounded, symmetric, positive definite operator over the set of a symmetric matrices in $\mathbb{R}^{d \times d}$:

$$\mathcal{A}\underline{v} = c_1 \underline{v} + c_2 \text{trace}(\underline{v}) \underline{I} \quad (5.3)$$

In Eq. (5.3), \underline{v} is a symmetric second order tensor in $\mathbb{R}^{d \times d}$.

In the hypothesis of homogeneous, isotropic, linear materials, we have that

$$c_1 = \frac{1}{2\mu}, \quad c_2 = \frac{-\lambda}{2\mu(3\lambda + 2\mu)}, \quad (5.4)$$

where μ, λ are the Lamé parameters [136]. Moreover we assume that the volumetric force term \underline{F}_{ext} lies in $\mathbf{L}^2(\Omega \times (0, T))$, $\underline{g}_D \in \mathbf{H}^{1/2}(\Gamma_D) = [H^{1/2}(\Gamma_D)]^d$, $\underline{g}_N \in \mathbf{L}^2(\Gamma_N) = [L^2(\Gamma_N)]^d$, and $\underline{F}_{target} \in \mathbf{L}^2(\Gamma_I) = [L^2(\Gamma_I)]^d$.

With these premises we can write the variational formulation of the problem: we look for $\underline{\underline{\sigma}} \in \mathbf{S}(\Omega) = \{\underline{\underline{S}} \in \mathbb{R}^{d \times d} : \underline{\underline{S}} = \underline{\underline{S}}^T\}$, $\underline{u} \in \mathbf{L}^2(\Omega)$, $\hat{\underline{u}} \in \text{span}\langle \underline{\underline{\varphi}} \rangle \oplus H_{00}^{1/2}(\Gamma_N)^d$, where

$$H_{00}^{1/2}[(\Gamma_N)]^d = \{\underline{\underline{\varphi}} \in [H^{1/2}(\Gamma)]^d : \underline{\underline{\varphi}} = \underline{0} \text{ on } \Gamma_D \cup \Gamma_I\},$$

such that for all $\underline{v} \in \mathbf{S}(\Omega)$, $\underline{w} \in \mathbf{L}^2(\Omega)$, $\underline{\mu} \in \text{span}\langle \underline{\underline{\varphi}} \rangle \oplus [H_{00}^{1/2}(\Gamma_N)]^d$ we have

$$(\mathcal{A}\underline{\underline{\sigma}}, \underline{v})_\Omega - (\underline{\underline{\varepsilon}}(\underline{u}), \underline{v})_\Omega = \underline{0}, \quad (5.5a)$$

$$- (\nabla \cdot \underline{\underline{\sigma}}, \underline{w})_\Omega + \left(\rho \frac{\partial^2 \underline{u}}{\partial t^2}, \underline{w} \right)_\Omega = (\underline{F}_{ext}, \underline{w})_\Omega, \quad (5.5b)$$

$$\langle \underline{\underline{\sigma}} \cdot \underline{n}, \underline{\underline{\mu}} \rangle_{\Gamma_N \cup \Gamma_I} = \langle \underline{g}_N, \underline{\underline{\mu}} \rangle_{\Gamma_N} + \underline{F}_{target} |\Gamma_I|^{-1} \langle \underline{\underline{\mu}}, \underline{1} \rangle_{\Gamma_I}, \quad (5.5c)$$

$$\langle \underline{\hat{u}}, \underline{\underline{\mu}} \rangle_{\Gamma_D} = \langle \underline{g}_D, \underline{\underline{\mu}} \rangle_{\Gamma_D}. \quad (5.5d)$$

After defining the variational formulation, we would like to give some insights to enlighten the motivation of constraint (5.2d). First, we multiply (5.1b) by $\frac{\partial u}{\partial t}$ and we integrate it on the whole domain Ω :

$$\rho \int_{\Omega} \frac{\partial^2 u}{\partial t^2} \frac{\partial u}{\partial t} d\Omega = \int_{\Omega} \nabla \cdot \underline{\underline{\sigma}} \frac{\partial u}{\partial t} d\Omega + \int_{\Omega} \underline{F}_{ext} \frac{\partial u}{\partial t} d\Omega$$

Denoting by $\mathcal{F} := \int_{\Omega} \underline{F}_{ext} \frac{\partial u}{\partial t} d\Omega$, we can write:

$$\frac{\rho}{2} \frac{d}{dt} \int_{\Omega} \left| \frac{\partial u}{\partial t} \right|^2 = \int_{\Omega} \nabla \cdot \left(\underline{\underline{\sigma}} \frac{\partial u}{\partial t} \right) d\Omega - \int_{\Omega} \underline{\underline{\sigma}} : \nabla \left(\frac{\partial u}{\partial t} \right) d\Omega + \mathcal{F} \quad (5.6a)$$

$$\frac{\rho}{2} \frac{d}{dt} \left\| \frac{\partial u}{\partial t} \right\|_{\mathbf{L}^2(\Omega)}^2 = \int_{\partial\Omega} \frac{\partial u}{\partial t} \underline{\underline{\sigma}} \underline{n} d\partial\Omega - \int_{\Omega} \underline{\underline{\sigma}} : \nabla \left(\frac{\partial u}{\partial t} \right) d\Omega + \mathcal{F} \quad (5.6b)$$

Next, we rewrite

$$\int_{\Omega} \underline{\underline{\sigma}} : \nabla \left(\frac{\partial u}{\partial t} \right) d\Omega = \int_{\Omega} \lambda (\nabla \cdot \underline{u}) \underline{I} : \nabla \left(\frac{\partial u}{\partial t} \right) d\Omega + \int_{\Omega} 2\mu \underline{\underline{\varepsilon}}(\underline{u}) : \nabla \left(\frac{\partial u}{\partial t} \right) d\Omega \quad (5.7a)$$

$$= \lambda \int_{\Omega} (\nabla \cdot \underline{u}) : \frac{\partial}{\partial t} (\nabla \underline{u}) d\Omega + 2\mu \int_{\Omega} \underline{\underline{\varepsilon}}(\underline{u}) : \frac{\partial}{\partial t} (\nabla \underline{u}) d\Omega \quad (5.7b)$$

$$= \lambda \int_{\Omega} (\nabla \cdot \underline{u}) \frac{\partial}{\partial t} (\nabla \cdot \underline{u}) d\Omega + 2\mu \int_{\Omega} \underline{\underline{\varepsilon}}(\underline{u}) : \frac{\partial}{\partial t} \underline{\underline{\varepsilon}}(\underline{u}) d\Omega \quad (5.7c)$$

$$= \frac{\lambda}{2} \frac{d}{dt} \|\nabla \cdot \underline{u}\|_{\mathbf{L}^2(\Omega)}^2 + \mu \frac{d}{dt} \|\underline{\underline{\varepsilon}}(\underline{u})\|_{\mathbf{L}^2(\Omega)}^2 \quad (5.7d)$$

$$(5.7e)$$

and finally obtain by substitution of (5.7e) into (5.6b):

$$\frac{\rho}{2} \frac{d}{dt} \left\| \frac{\partial u}{\partial t} \right\|_{\mathbf{L}^2(\Omega)}^2 + \frac{\lambda}{2} \frac{d}{dt} \|\nabla \cdot \underline{u}\|_{\mathbf{L}^2(\Omega)}^2 + \mu \frac{d}{dt} \|\underline{\underline{\varepsilon}}(\underline{u})\|_{\mathbf{L}^2(\Omega)}^2 = \int_{\partial\Omega} \frac{\partial u}{\partial t} \underline{\underline{\sigma}} \underline{n} d(\partial\Omega) + \mathcal{F}$$

Our interest is now focused on

$$\int_{\partial\Omega} \frac{\partial u}{\partial t} \underline{\underline{\sigma}} \underline{n} d(\partial\Omega).$$

We can distinguish three different situations:

1. **Neumann BC** on Γ_N : $\underline{\underline{\sigma}} \underline{n} = \underline{g}_N(\underline{x}, t) \implies$ we incorporate $\int_{\Gamma_N} \frac{\partial u}{\partial t} \underline{g}_N d\Gamma_N$

2. **Dirichlet** on Γ_D : $\underline{u} = \underline{g}_D(\underline{x}, t) \implies$ we obtain $\int_{\Gamma_D} \frac{\partial \underline{g}_D}{\partial t} \underline{\underline{\sigma}} \underline{n} d\Gamma_D$

3. **IBC** on Γ_I : we have $\int_{\Gamma_I} \underline{\underline{\sigma}} \underline{n} d\Gamma_I = \underline{F}_{target}$, but $\int_{\Gamma_I} \frac{\partial u}{\partial t} \underline{\underline{\sigma}} \underline{n} d\Gamma_I$ is still undefined.

Note that only if

$$\frac{\partial \underline{u}}{\partial t} \Big|_{\Gamma_I} = \frac{\partial \underline{u}}{\partial t}(t) \quad (5.8)$$

we can write $\int_{\Gamma_I} \frac{\partial \underline{u}}{\partial t} \underline{\sigma} \cdot \underline{n} \, d(\partial\Omega) = \frac{\partial \underline{u}}{\partial t} \int_{\Gamma_I} \underline{\sigma} \cdot \underline{n} \, d(\partial\Omega) = \frac{\partial \underline{u}}{\partial t} \underline{F}_{target}$ on Γ_I . Thus, Eq. (5.8) leads to the supplementary constraint on the integral boundary condition specified in Eq. (5.2d).

Remark 1. In the OMVS we use only Dirichlet and Neumann BCs for the linear elasticity system, whereas we employ the IBC in an application proposed in Ch. 24.

2 Poroelasticity

We will describe the lamina cribrosa as a poroelastic medium, where hemodynamics and biomechanics are combined to model the physiological behavior of this tissue, with the goal of understanding the interactions between these two biological mechanisms. In order to achieve this scope we started from the poroelastic system introduced by Biot [15] and subsequently studied by [58, 52, 249, 151]:

$$\begin{cases} \rho \frac{\partial^2 \underline{u}}{\partial t^2} - \nabla \cdot (\underline{\sigma}_{el} - \alpha p \underline{I}) = \underline{F}_{el} & \text{in } \Omega \times [0, T] \\ \frac{\partial \zeta}{\partial t} + \nabla \cdot \underline{j} = F_{fl} & \text{in } \Omega \times [0, T] \end{cases} \quad (5.9)$$

where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ is the computational domain, \underline{u} is the solid displacement, $\underline{\sigma}_{el}$ is elastic stress tensor, \underline{F}_{el} is the volumetric force term, p is the fluid pressure, α is the Biot coefficient, \underline{j} is the discharge velocity, F_{fl} is the volumetric fluid source term, and ζ is the fluid content. The definition of ζ , $\underline{\sigma}$, and \underline{j} , reads:

$$\zeta = \frac{1}{M} p + \alpha \nabla \cdot \underline{u} \quad (5.10)$$

$$\underline{\sigma}_{el} = \mu (\nabla \underline{u} + \nabla^T \underline{u}) + \lambda (\nabla \cdot \underline{u}) \underline{I} \quad (5.11)$$

$$\underline{j} = -\underline{\mathcal{K}} \nabla p \quad (5.12)$$

where M is the Biot modulus, μ and λ are the Lamé parameters of the elastic matrix, and $\underline{\mathcal{K}}$ is the permeability tensor. Exploiting these definitions we can rewrite the system (5.9) as:

$$\begin{cases} \rho \frac{\partial^2 \underline{u}}{\partial t^2} - (\lambda + \mu) \nabla (\nabla \cdot \underline{u}) - \mu \Delta \underline{u} + \alpha \nabla p = \underline{F}_{el} & \text{in } \Omega \times [0, T] \\ \frac{1}{M} \frac{\partial p}{\partial t} + \alpha \nabla \cdot \frac{\partial \underline{u}}{\partial t} - \nabla \cdot \underline{\mathcal{K}} \nabla p = F_{fl} & \text{in } \Omega \times [0, T] \end{cases}$$

From this formulation, we introduce the hypothesis that the exchange of fluid content is only due to p , i.e. $\alpha \frac{\partial}{\partial t} (\nabla \cdot \underline{u}) \approx 0$. Thus, we obtain the following simplified system of PDEs:

$$\begin{cases} \rho \frac{\partial^2 \underline{u}}{\partial t^2} - \nabla \cdot \underline{\sigma} = \underline{F}_{el} & \text{in } \Omega \times [0, T] \\ \frac{1}{M} \frac{\partial p}{\partial t} + \nabla \cdot \underline{j} = F_{fl} & \text{in } \Omega \times [0, T] \end{cases} \quad (5.13)$$

where $\underline{\sigma} = \underline{\sigma}_{el} - \alpha p \underline{I}$ is the stress tensor of the mixture or total stress. System (5.13) is accompanied by appropriate boundary and initial conditions.

Remark 2. The well posedness of the poroelastic system (5.9) is proved by Showalter in [222].

3 Poroelastic model for the lamina cribrosa

For our application we are interested in using the poroelastic system presented in the previous section for modeling the lamina cribrosa. The idea is to use the hemodynamic framework of *Level 1* combined with the biomechanical description *via* the model proposed in (5.13).

In the context of the lamina cribrosa there are more advanced studied in specific parts of this model, for instance the use of a nonlinear model to describe the biomechanics of the lamina [202], or the use of a poro-viscoelastic model to characterize the interaction between the blood flow and the elastic tissue within the optic nerve head [184]. The novelty that we are bringing, despite the hypothesis introduced and the use of a simple linear elastic model, is the conception of this framework in which the 3D model of the lamina cribrosa does not have artificial boundary conditions but *via* an IBC it is coupled with a system of ODEs representing the blood flow circulation. Moreover, the use of Eqs. (5.13) has to be seen as a first step to create the "big picture" of the ocular system in which each mathematical component can be replaced with more sophisticated and adapted models as soon as the research evolves.

We sketch the new scheme for *Level 2a* in Fig. 5.1.

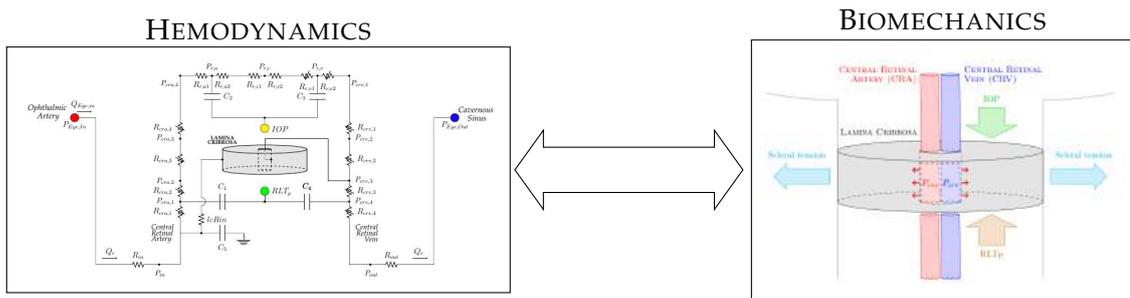


Figure 5.1 – *Level 2a* scheme.

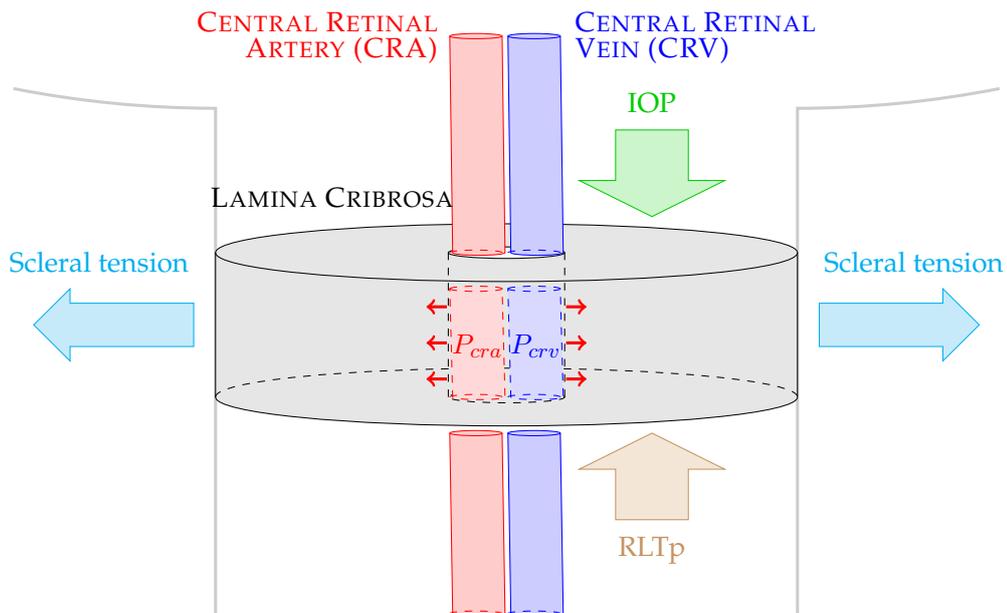


Figure 5.2 – Biomechanical boundary condition for the lamina cribrosa.

The data used for the hemodynamics part of the model are the same as in Ch. 4.

Remark 3. *Hereafter we do not consider external volumetric forces or volumetric fluid source terms, i.e. $\underline{F}_{el} = \underline{0}$ and $F_{fl} = 0$.*

For what concerns the biomechanics, the BCs are depicted in Fig. 5.2, i.e. we have:

- $\underline{\sigma}_{el} \cdot \underline{n} = -p_{cra} \underline{n}$ on the internal boundary of the lamina; in this case - due to lack of biological information - we have assumed that all the internal boundaries feel the pressure within the CRA that is compressing (negative sign), whereas from a strictly physiological viewpoint there is a subset that is influenced by the CRV; thus, we are overestimating this pressure since $p_{cra} > p_{crv} \forall t \in [0, T]$;
- $\underline{\sigma} \cdot \underline{n} = -IOP \underline{n}$ on the top boundary; the negative sign is to indicate the fact that IOP is compressing the tissue;
- $\underline{\sigma} \cdot \underline{n} = -RLTp \underline{n}$ on the bottom boundary; also in this case the negative sign is to indicate the compressing effect of the CSF behind the eye on the tissue;
- $\underline{u} = \underline{0}$ on the lateral boundary; at first instance we did not consider the effect of the sclera on the optic nerve head; this hypothesis will be removed in the next section when we add the biomechanical description of the sclera.

The baseline values for the BCs for *Level 2a* are $p_{cra} = 80$ mmHg, $IOP = 15$ mmHg, and $RLTp = 7$ mmHg. The values for the Lamé coefficients to describe the elastic properties of the lamina cribrosa are summarized in Tab. 5.1 and have been retrieved from [35] and [253] in case of small stresses (0.0 – 8.0 kPa), which include the physiological range of pressures involved in the realistic framework.

LAMÉ PARAMETER	VALUE	UNIT
μ	0.12	MPa
λ	5.88	MPa

Table 5.1 – Elastic properties of the lamina cribrosa for *Level 2a*.

4 Biomechanics of sclera, choroid, retina and cornea

In this section we extend the previous model introducing the biomechanics of the sclera (Fig. 5.3a), choroid (Fig. 5.3b), retina (Fig. 5.3c) and cornea (Fig. 5.3d). This is the current most advanced version of the OMVS and it is called *Level 2b*.

The hemodynamics model is the same as *Level 2a* (Fig. 4.1), which means that we account for the vasculature in the back of the eye thanks to the 0D model described in Ch. 3 and the blood perfusion of the lamina cribrosa *via* a 3D poroelastic model presented in Ch. 4. For what concerns the biomechanics we impose the continuity of stresses and displacements on the interfaces between the different tissues:

$$[\underline{\sigma}]_{\Gamma_{int}^-} = [\underline{\sigma}]_{\Gamma_{int}^+}$$

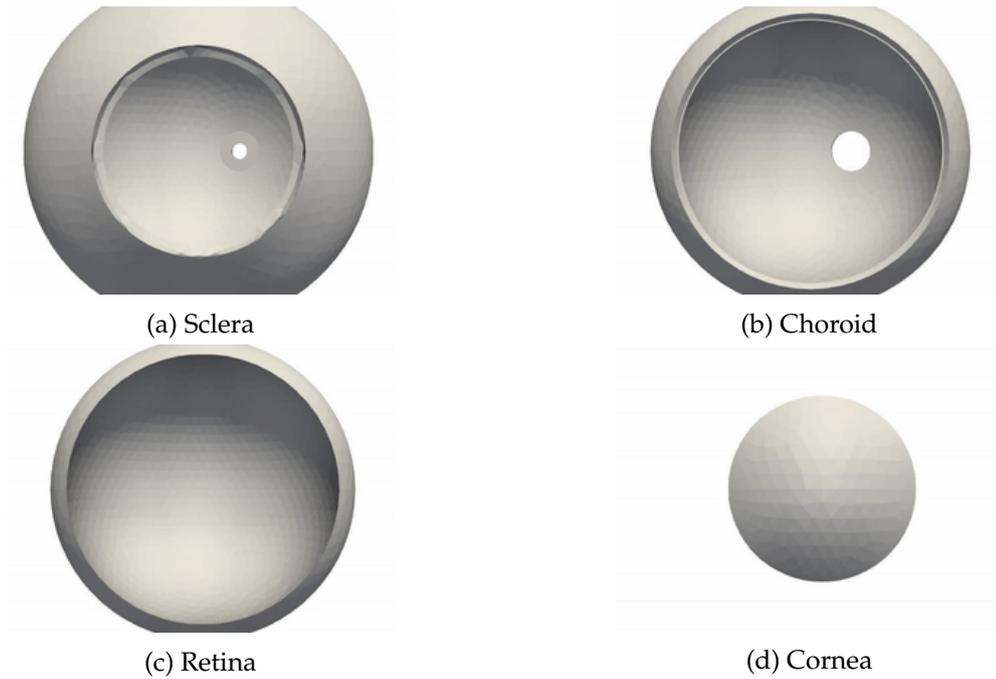


Figure 5.3 – Ocular tissues introduced in the biomechanical description of *Level 2b*.

$$[u]_{\Gamma_{int}^-} = [u]_{\Gamma_{int}^+}$$

where Γ_{int}^- and Γ_{int}^+ are the left and right interface of the shared internal faces of the ocular domain. Tab. 5.2 lists all the BCs for the linear elasticity equations solved within the biomechanical domain and the values of the data used; the different panels in Figs. 5.4 and 5.5 illustrate the labels for the name of the different boundaries in the complex 3D domain of *Level 2b*.

In Tab. 5.3, we resume the Lamé coefficients of each simulated ocular tissue for *Level 2b* model, values that have been retrieved from [75, 67].

Name boundary	BC type	Parameter	Value	Unit
SCLERA				
<i>Iris_Sclera</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -IOP \underline{n}$	<i>IOP</i>	15	mmHg
<i>Sclera_internalBC</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -IOP \underline{n}$	<i>IOP</i>	15	mmHg
<i>Pia_Sclera</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
<i>OpticNerve_Sclera</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
<i>Sclera_externalBC</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
CHOROID				
<i>BC_Choroid</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -IOP \underline{n}$	<i>IOP</i>	15	mmHg
<i>Choroid_Iris</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
RETINA				
<i>BC_spherical_Retina</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -IOP \underline{n}$	<i>IOP</i>	15	mmHg
<i>OpticNerve_Retina</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
CORNEA				
<i>Cornea_internalBC</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -IOP \underline{n}$	<i>IOP</i>	15	mmHg
<i>Cornea_externalBC</i>	$\underline{u} = \underline{0}$	[-]	[-]	[-]
LAMINA CRIBROSA				
<i>Out</i>	$\underline{\underline{\sigma}} \cdot \underline{n} = -RLTp \underline{n}$	<i>RLTp</i>	7	mmHg
<i>Hole</i>	$\underline{\underline{\sigma}}_{el} \cdot \underline{n} = -p_{cra} \underline{n}$	<i>p_{cra}</i>	80	mmHg

Table 5.2 – BCs data for *Level 2b* of the OMVS.

	LAMINA CRIBROSA		SCLERA		CHOROID		RETINA		CORNEA	
	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ
Value [MPa]	0.12	5.88	0.676	16.216	0.203	4.86	0.067	1.62	0.845	20.27
Reference	[35]	[253]	[75]	[174] [253]	[75]		[121]	[252]	[183]	[253]

Table 5.3 – Lamé parameters for *Level 2b* of the OMVS

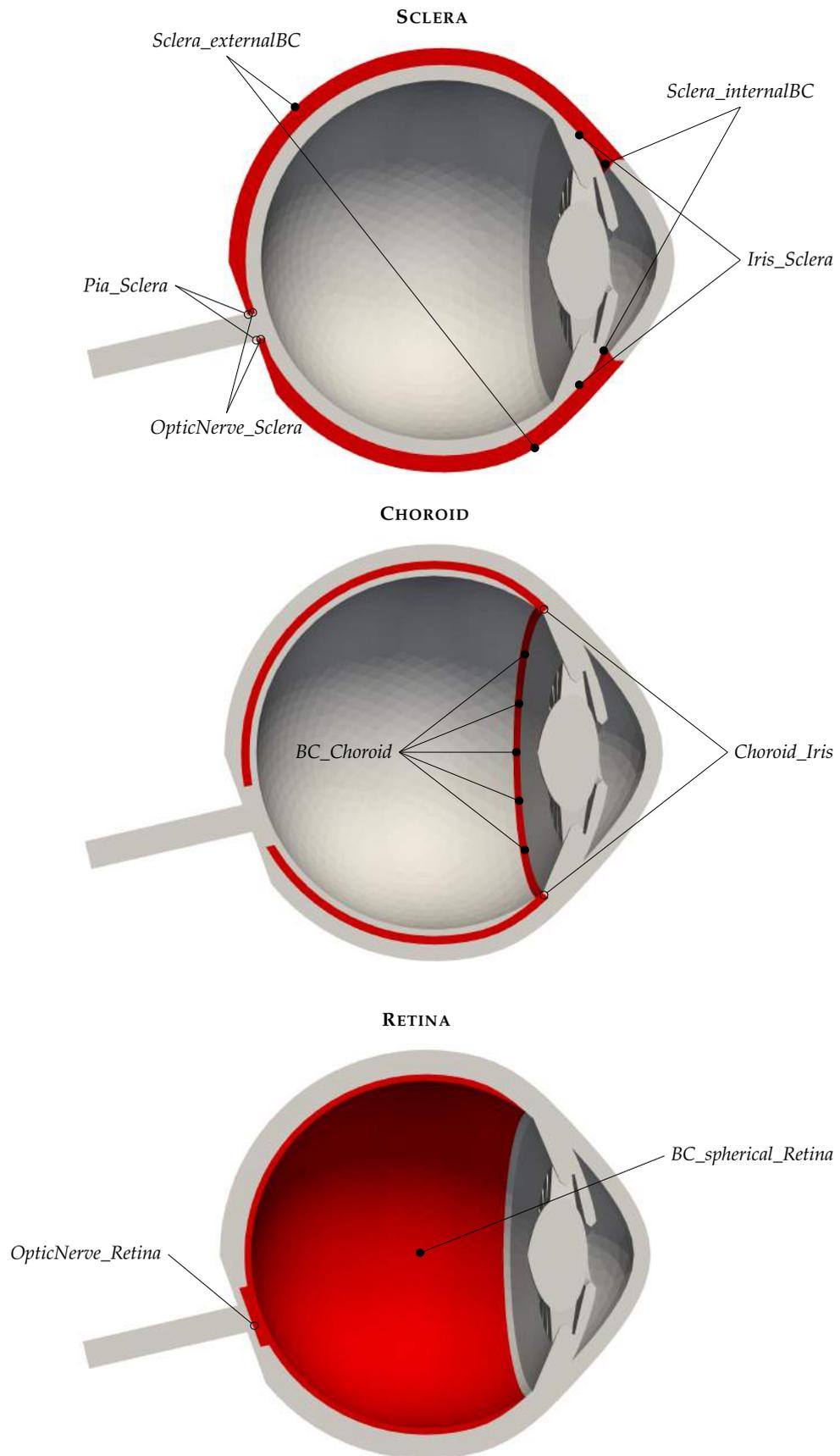


Figure 5.4 – BCs labels of the OMVS geometry. The original geometry has been clipped along the z-plane in order to view within the eye. For each subfigure, the domain of interest is highlighted in red.

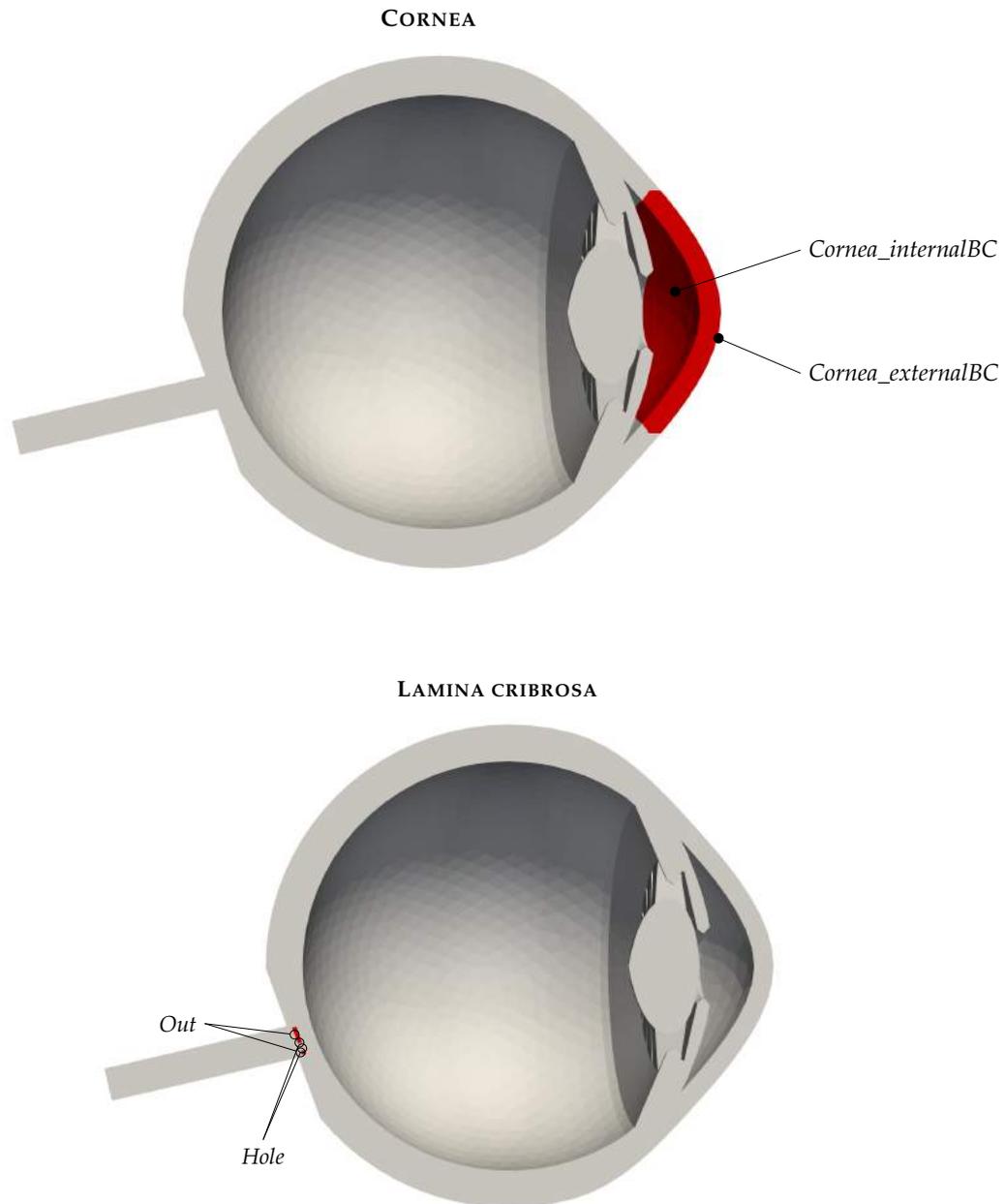


Figure 5.5 – BCs labels of the OMVS geometry. The original geometry has been clipped along the z-plane in order to view within the eye. For each subfigure, the domain of interest is highlighted in red.

Part III

Numerical methods

In this part we introduce the numerical methods adopted in order to solve the models presented in the previous part. Since we have adopted a Finite Element Method (FEM) approach, we present:

- the computational mesh, in particular the process needed to handle the complex ocular geometry;
- the time discretization, in particular a novel operator splitting strategy used to couple the 3D system of PDEs with the nonlinear system of ODEs;
- the spatial discretization, in particular the Hybridizable Discontinuous Galerkin (HDG) method with original non-standard boundary condition, namely Integral Boundary Condition (IBC).

Contents

1	Geometry	52
2	Mesh	54
3	Further comments	56

In this chapter we present the process we accomplished to build the geometry and the computational mesh used in the 3D simulations for *Level 1* and *Level 2* of the OMVS (Chs. 4 and 5). This work has been realized in collaboration with Christophe Trophime, a research engineer of the LCNMI (*Laboratoire National des Champs Magnétiques Intenses*) in Grenoble during CEMRACS (Centre d'Été Mathématique de Recherche Avancée en Calcul Scientifique) 2016.

The generation of the geometry and the mesh is realized thank to the software Salome [204], which is an integration platform for numerical simulations with pre- and post-processing tools. We are interested in the CAD (Computer Aided Design) module, which is based on OpenCascade[215] and its constructive solid geometry features, and it is also equipped with a Python script interface.

In the following sections we describe the overall process (Fig. 6.1) to obtain a computational mesh of the eye starting from a CAD image reporting, whenever possible, the python scripts used.

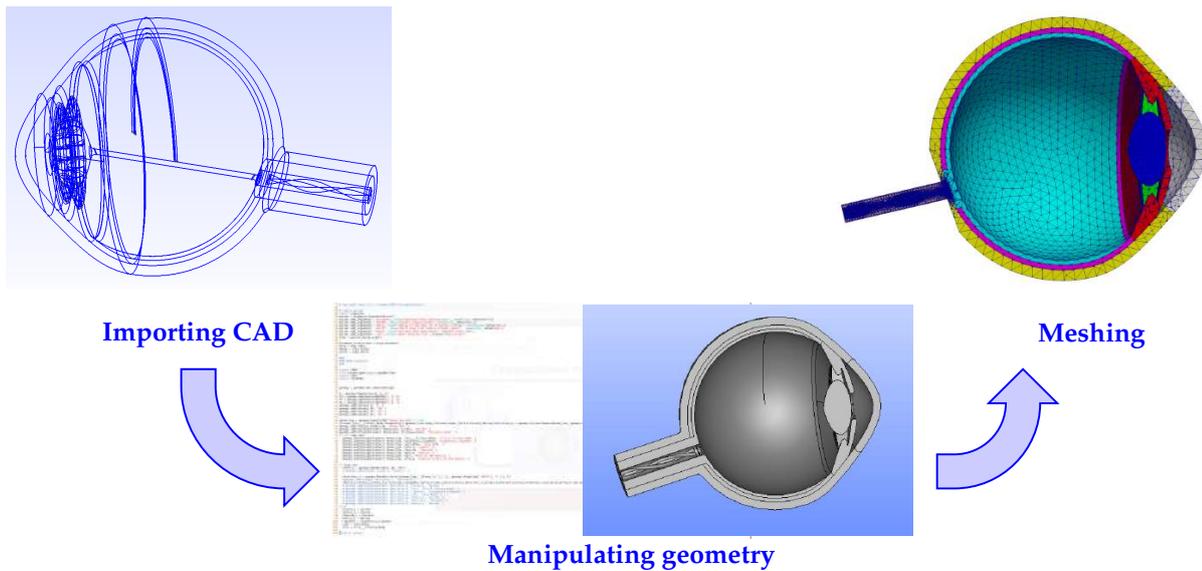


Figure 6.1 – Process to generate a computational mesh from a CAD drawing using the Salome platform.

1 Geometry

The starting CAD image of the eye is illustrated by Fig. 6.2 (courtesy of CEMOSIS[†]). This preliminary image is imported on Salome, where the following 3D ocular structures are identified: cornea, vitreous humor, iris and ciliary body, ligament, lens, sclera, choroid, retina, central retinal vein and artery.

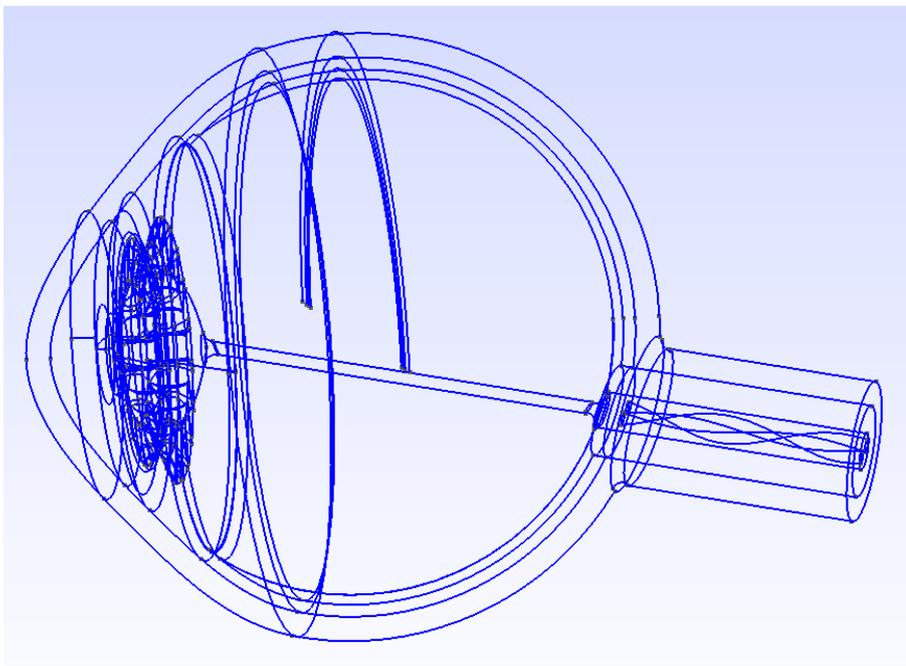


Figure 6.2 – CAD starting image. Courtesy of CEMOSIS[†].

Code 6.1 – Import CAD

[†]Center for MOdeling and SIMulation in Strasbourg. <http://www.cemosis.fr/>

```

1 Human_Eye = geompy.ImportSTEP("Human_Eye.stp", True)
2 [Cornea,Iris____Ciliary_Body,Suspensory_Ligament,Lens_Body,Vitreous_humor,Sclera,Choroid,Retina,
3   Vein,Artery] = geompy.ExtractShapes(Human_Eye, geompy.ShapeType["SOLID"], True)
4 geompy.addToStudy( Human_Eye, 'Human Eye' )
5 geompy.addToStudyInFather( Human_Eye, Cornea, 'Cornea' )
6 geompy.addToStudyInFather( Human_Eye, Vitreous_humor, 'Vitreous humor' )
7 geompy.addToStudyInFather( Human_Eye, Iris____Ciliary_Body, 'Iris & Ciliary Body' )
8 geompy.addToStudyInFather( Human_Eye, Suspensory_Ligament, 'Suspensory Ligament' )
9 geompy.addToStudyInFather( Human_Eye, Lens_Body, 'Lens Body' )
10 geompy.addToStudyInFather( Human_Eye, Sclera, 'Sclera' )
11 geompy.addToStudyInFather( Human_Eye, Choroid, 'Choroid' )
12 geompy.addToStudyInFather( Human_Eye, Retina, 'Retina' )
13 geompy.addToStudyInFather( Human_Eye, Vein, 'Vein of the Retina' )
14 geompy.addToStudyInFather( Human_Eye, Artery, 'Central Artery of the Retian' )

```

The code to import the STEP file and recover the architecture of the eye is detailed in Code 6.1.

The main part that we had to add to the starting CAD description was the lamina cribrosa, which is actually the crucial tissue we are interested in. Following the ocular anatomy presented in Ch. 2, we create a lamina cribrosa geometry (Fig. 6.3a) within the imported ocular framework (see Code 6.2). In the generation of this tissue, we made some geometrical parameters accessible and tunable of the lamina cribrosa by the user (see Tab. 6.1).

Code 6.2 – Build Lamina Cribrosa

```

1 [Face_18,Face_19,Face_20,Face_21,Face_22,Face_23,Face_24,Face_25,Face_26,Face_27,Face_28,
2   Face_29,Face_30] = geompy.ExtractShapes(Retina, geompy.ShapeType["FACE"], True)
3 [Face_1,Face_2,Face_3,Face_4,Face_5,Face_6,Face_7,Face_8,Face_9,Face_10,Face_11,Face_12,
4   Face_13,Face_14,Face_15,Face_16,Face_17] =
5   geompy.ExtractShapes(Sclera, geompy.ShapeType["FACE"], True)
6 [Wire_7] = geompy.ExtractShapes(Face_13, geompy.ShapeType["WIRE"], True)
7 [Wire_8] = geompy.ExtractShapes(Face_14, geompy.ShapeType["WIRE"], True)
8 [Wire_5] = geompy.ExtractShapes(Face_15, geompy.ShapeType["WIRE"], True)
9 [Wire_6] = geompy.ExtractShapes(Face_16, geompy.ShapeType["WIRE"], True)
10 [Wire_3] = geompy.ExtractShapes(Face_26, geompy.ShapeType["WIRE"], True)
11 [Wire_4] = geompy.ExtractShapes(Face_27, geompy.ShapeType["WIRE"], True)
12 [Edge_1,Edge_2,Edge_3,Edge_4] = geompy.ExtractShapes(Wire_3,
13   geompy.ShapeType["EDGE"], True)
14 [Edge_5,Edge_6,Edge_7,Edge_8] = geompy.ExtractShapes(Wire_4,
15   geompy.ShapeType["EDGE"], True)
16 Base_retina = geompy.MakeFuseList([Edge_5, Edge_1], True, True)
17 Base_R = geompy.MakeCDG(Base_retina)
18 Top_Retina = geompy.MakeFuseList([Edge_8, Edge_4], True, True)
19 Top_R = geompy.MakeCDG(Top_Retina)
20 N_Base_retina = geompy.MakeVector(Base_R, Top_R)
21
22 # Should get vector in the Base_retina plane
23 [P0, P1] = geompy.ExtractShapes(Edge_5, geompy.ShapeType["VERTEX"], True)
24 T0_Base_retina = geompy.MakeVector(Base_R, P0)
25 T1_Base_retina = geompy.CrossProduct(N_Base_retina, T0_Base_retina)
26
27 [Edge_9,Edge_10,Edge_11,Edge_12,Edge_13]
28   = geompy.ExtractShapes(Wire_5, geompy.ShapeType["EDGE"], True)
29 [Edge_14,Edge_15,Edge_16,Edge_17,Edge_18]
30   = geompy.ExtractShapes(Wire_6, geompy.ShapeType["EDGE"], True)
31 [Edge_19,Edge_20,Edge_21,Edge_22] = geompy.ExtractShapes(Wire_7, geompy.ShapeType["EDGE"], True)
32 [Edge_23,Edge_24,Edge_25,Edge_26] = geompy.ExtractShapes(Wire_8, geompy.ShapeType["EDGE"], True)
33 Inner_Bas_sclera = geompy.MakeFuseList([Edge_19, Edge_23], True, True)
34 Inner_Bas_S = geompy.MakeCDG(Inner_Bas_sclera)
35 Vector_2 = geompy.MakeVector(Base_R, Inner_Bas_S)
36 Lamina_cribrosa_Base = geompy.MakeFaceWires([Inner_Bas_sclera], 1)
37 Translation_2 = geompy.MakeTranslationVectorDistance(Base_retina, Vector_2, distance_from_sclera)
38 Lamina_Base = geompy.MakeFaceWires([Translation_2], 1)
39
40 [Wire_9] = geompy.ExtractShapes(Lamina_Base, geompy.ShapeType["WIRE"], True)
41 Laminia_Top = geompy.MakeTranslationVectorDistance(Lamina_Base, Vector_2, depth)
42 [Wire_10] = geompy.ExtractShapes(Laminia_Top, geompy.ShapeType["WIRE"], True)
43 G_Lamina_T = geompy.MakeCDG(Laminia_Top)
44 Vector_3 = geompy.MakeVector(Inner_Bas_S, G_Lamina_T)
45
46 Scaled_Lamina_w_hole_1 = None
47 try:
48   DZ_Hole = geompy.BasicProperties(N_Base_retina)[0]
49   DR_Hole = geompy.BasicProperties(T0_Base_retina)[0]
50   # should check if shift is less than DR_Hole - 2*hole
51   if shift > DR_Hole - 2*hole:
52     print "pb with shift size"
53
54   Base_R_1 = geompy.MakeTranslationVectorDistance(Base_R, N_Base_retina, -0.5*DZ_Hole)
55   Cylinder_1 = geompy.MakeCylinder(Base_R_1, N_Base_retina, hole, 1.5*DZ_Hole)
56   Translation_Cyl = geompy.MakeTranslationVectorDistance(Cylinder_1, T0_Base_retina, shift)

```

```

57
58 G_Lamina_Base = geompy.MakeCDG(Lamina_Base)
59 Scaled_Lamina_Base = geompy.MakeScaleTransform(Lamina_Base, G_Lamina_Base, 1.2)
60 Scaled_Lamina_cribosa = geompy.MakePipe(Scaled_Lamina_Base, Vector_3)
61
62 Scaled_Lamina_w_hole = geompy.MakeCutList(Scaled_Lamina_cribosa, [Translation_Cyl], True)
63 Scaled_Lamina_w_hole_1 = geompy.MakeCutList(Scaled_Lamina_w_hole, [Retina], True)
64
65 except:
66     print "Failed to build Lamina"

```

Consequently, the other tissues (*i.e.*:retina, sclera, optic nerve ...) were modified and adjusted in order to fit the new structure. A cut of the new geometry of the eye built using Salome is displayed in Fig. 6.3b. In Tab. 6.1 we recap the input parameters that are currently tunable by the user to obtain a patient-specific geometry.

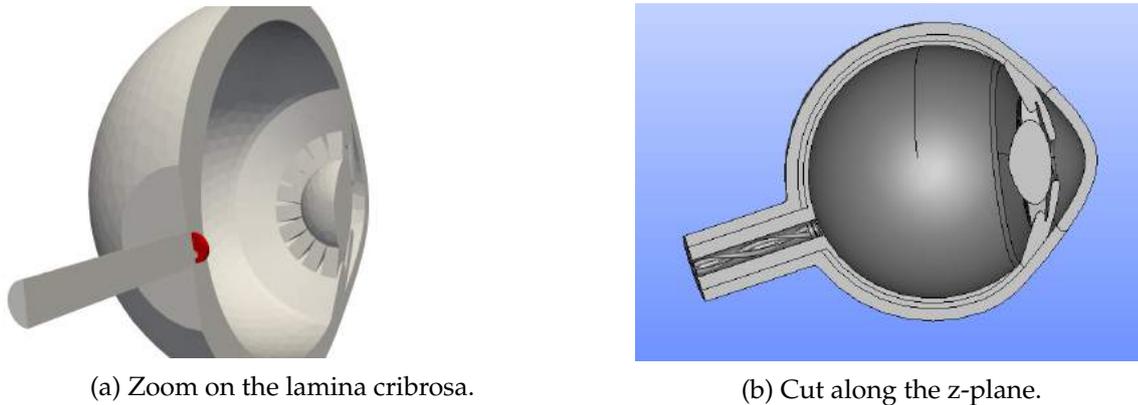


Figure 6.3 – Geometry of the eye realized with Salome.

PARAMETER	DEFAULT VALUE	DESCRIPTION
hsize_lamina	0.05	max size of the mesh grid h for the lamina cribrosa sub-domain
hsize_eye	1.0	max size of the mesh grid h for all the eye except the lamina cribrosa sub-domain
distance	0.25 mm	lamina cribrosa distance from retina/sclera
width	0.2 mm	lamina cribrosa width
hole	0.2 mm	radius of the lamina cribrosa hole
shift	0.3 mm	hole shift from lamina cribrosa center
eye_length	26.1 mm	ocular axial length
corneal_thickness	1.0	corneal thickness proportion with respect to the original CAD thickness

Table 6.1 – Geometrical parameters that can be modified by the user during the generation of the computational mesh.

2 Mesh

The mesh module in Salome is used to generate 1D/2D/3D meshes. The following meshing plug-ins are available:

free: NETGEN, Gmsh;

commercial: MeshGems CADSurf, MeshGems Tetra, MeshGems Tetra_HPC, MeshGems Hexa, MeshGems Hybrid.

We set markers on every volume or surface of the structured mesh allowing simulations that can be run only on single parts of the eye or on the whole geometry (see Code 6.3).

Code 6.3 – Set mesh markers

```

1  ## Set Markers
2  Cornea_3 = Mesh_1.GroupOnGeom(Cornea_2,'Cornea',SMESH.VOLUME)
3  Iris_3 = Mesh_1.GroupOnGeom(Iris_2,'Iris',SMESH.VOLUME)
4  Ligament_3 = Mesh_1.GroupOnGeom(Ligament_2,'Ligament',SMESH.VOLUME)
5  Lens_3 = Mesh_1.GroupOnGeom(Lens_2,'Lens',SMESH.VOLUME)
6  Sclera_3 = Mesh_1.GroupOnGeom(Sclera_2,'Sclera',SMESH.VOLUME)
7  Choroid_3 = Mesh_1.GroupOnGeom(Choroid_2,'Choroid',SMESH.VOLUME)
8  Retina_3 = Mesh_1.GroupOnGeom(Retina_2,'Retina',SMESH.VOLUME)
9  Lamina_3 = Mesh_1.GroupOnGeom(Lamina_2,'Lamina',SMESH.VOLUME)
10 Vein_3 = Mesh_1.GroupOnGeom(Vein_2,'Vein',SMESH.VOLUME)
11 Artery_3 = Mesh_1.GroupOnGeom(Artery_2,'Artery',SMESH.VOLUME)
12 Pia_3 = Mesh_1.GroupOnGeom(Pia_2,'Pia',SMESH.VOLUME)
13 OpticNerve_3 = Mesh_1.GroupOnGeom(OpticNerve_2,'OpticNerve',SMESH.VOLUME)
14 print "Create Groups from Geometry"
15
16 Out_3 = Mesh_1.GroupOnGeom(Out,'Out',SMESH.FACE)
17 Out_3.SetColor( SALOMEDS.Color( 1, 0.666667, 0 ))
18 Hole_3 = Mesh_1.GroupOnGeom(Hole,'Hole',SMESH.FACE)
19 Hole_3.SetColor( SALOMEDS.Color( 1, 0.666667, 0 ))
20 Hole_3.SetName( 'Hole' )
21 print "Set Color for BC Groups on Lamina"
22
23 ## Set names of Mesh objects
24 smesh.SetName(Out_3, 'Out')
25 smesh.SetName(Hole_3, 'Hole')
26
27 for item in Others:
28     Name = item.GetName()
29     BC_Group_Mesh = Mesh_1.GroupOnGeom(item,item.GetName(),SMESH.FACE)
30     BC_Group_Mesh.SetColor( SALOMEDS.Color( 1, 0.666667, 0.333333 ))
31
32 print "Set specific BC Groups for Retina"
33 Done = ["Lamina"]
34 for interface in Interfaces:
35     for item in interface:
36         Name = item.GetName()
37         print Name
38         if not Name in Done:
39             Done.append(Name)
40             print "add %s"%Name
41             BC_Group_Mesh = Mesh_1.GroupOnGeom(item,Name,SMESH.FACE)
42             BC_Group_Mesh.SetColor( SALOMEDS.Color( 1, 0.666667, 0.333333 ))
43             print "Set Color for BC Groups on Retina"

```

Next, we used the NETGEN meshing plug-in [216] to specify the parameters of the mesh and eventually build the mesh (see Code 6.4).

Code 6.4 – Create and build the computational mesh

```

1  try:
2      # Create mesh for every solid
3      for solid in Solids:
4          Name = solid.GetName()
5          algo3d = Mesh_1.Tetrahedron(algo=smeshBuilder.NETGEN_1D2D3D, geom=solid)
6          if "Lamina" in Name:
7              lamina_params = algo3d.Parameters()
8              lamina_params.SetMaxSize(hsize_lamina)
9              lamina_params.SetMinSize(hsize_lamina/5)
10         else:
11             eye_params = algo3d.Parameters()
12             eye_params.SetMaxSize(hsize_eye)
13             smesh.SetName(algo3d, solid.GetName() )
14             print "Define NETGEN Hypothesis"
15     except:
16         print "Failed to define NETGEN hypothesis"
17
18     # Actually build Mesh
19     try:
20         isDone = Mesh_1.Compute()
21         Mesh_1.ExportMED( "eye.med", 0, SMESH.MED_V2_2, 1, None ,1)
22         print 'Mesh built successfully'
23
24     except:
25         print "Failed to create Mesh"

```

Fig. 6.4a shows the generated computational mesh for the lamina cribrosa, while Fig. 6.4b displays the cut along the z-plane of the overall computational mesh.

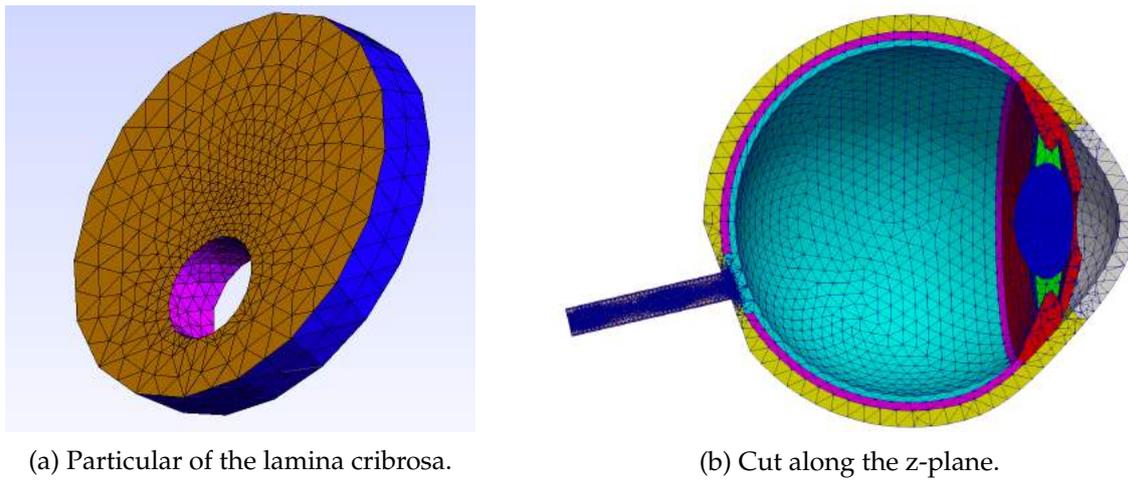


Figure 6.4 – Mesh of the eye realized with Salome using NETGEN algorithms.

3 Further comments

In order to compute the geometry and build the computational mesh using the default values, the code takes a total time of 272.6s of which 186.1 for the geometry and 84.7 for the mesh. Tab. 6.2. The algorithm used to build the mesh is very robust in terms of choices for the mesh grid sizes, it automatically selects the largest h with the constraint of the maximum specified by the user and without generating non-conforming elements.

HSIZE_LAMINA	HSIZE_EYE	NB OF ELEMENTS	GEOMETRY TIME [s]	MESH TIME [s]	TOTAL TIME [s]
0.05	1.0	336267	186.1306059	84.70704699	272.5534668
0.025	0.5	611801	186.7017241	143.439055	331.88304
0.1	2	288663	186.257086	75.53445411	263.531353
1	1	301017	188.449379	78.61220908	268.8120461
0.05	0.25	1184958	184.623606	297.347297	483.7180319

Table 6.2 – Computational times and data using different h sizes for the mesh realized with Salome.

In conclusion, we have developed from a CAD drawing a geometry of the whole eye, selecting some appropriate parameters that can be tuned, in order to generate a complex computational mesh. This mesh presents:

- 12 sub-domains;
- 26 boundaries / interfaces;
- conforming elements at edges and at interfaces;
- specific discretization for the lamina cribrosa due to its different geometrical dimensions with respect to the other components.

This generating process has addressed the goal of having a full realistic description of the ocular architecture, which is used by the OMVS for patient-specific numerical simulations.

3D-0D operator splitting method

Contents

1	Introductory notation	58
2	Energy of the system	60
3	Dimensional analysis	63
4	Operator splitting technique	64
5	Two-step splitting	65

One of the main issues in the context of mathematical models for biomedical applications is the choice of the region of interest and consequently of the boundary conditions (BCs). To avoid an artificial behavior due to bad selection of BCs (*e.g.* backward flow, spurious pressure oscillations), especially for hemodynamics, we decided to couple the region of interest, which is described with a three-dimensional model, with the characterization of the blood circulation in the back of the eye with a lumped parameter approach (see Fig. 7.1). In this way, we also incorporate the blood circulation in the retina without increasing the computational effort.

The challenge introduced by a 3D-0D coupling is mainly due to the instabilities of numerical methods or again the computational cost.

In the literature many different approaches have been proposed and several research teams have studied this topic, see for instance [191, 194, 74, 158] for a non-exhaustive view on the vast panorama of works on coupling techniques. For our purpose, we decided to adopt an operator splitting method inspired by [84, 30]. Note that this innovative coupling, however, is a time-splitting technique, thus, it doesn't take care of possible spatial multiscale issues. The HDG formulation that we will describe in the following chapter (Ch. 8) supports the direct solution of the integral boundary condition without any sub-iteration. This strategy allows us to compute at the same time the pressure on that boundary, obtaining therefore a natural spatial coupling between the 3D and 0D.

The operator splitting method presented in this chapter has been published in [93].

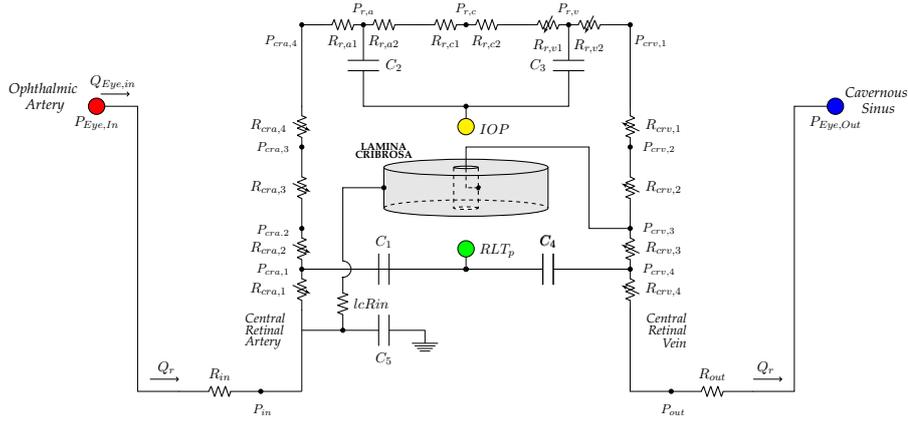


Figure 7.1 – Multiscale 3D-0D coupling between the retinal vasculature and the lamina cribrosa.

1 Introductory notation

With the aim of explaining the novel operator splitting method we propose, we extend our discussion to a more general framework, sketched in Fig. 7.2.

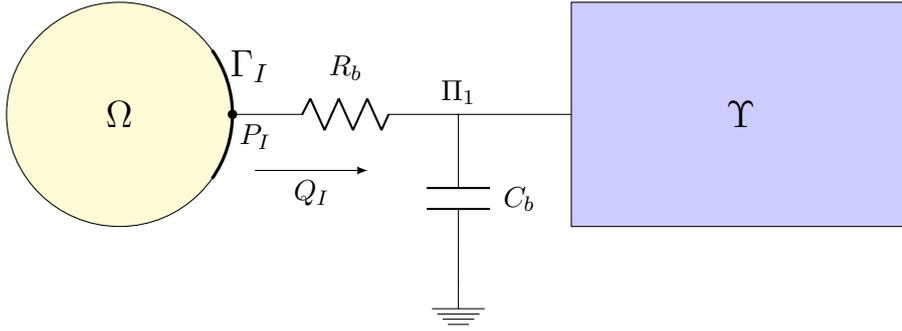


Figure 7.2 – Geometrical architecture of the coupled system; Ω corresponds to the region where the system of PDEs (7.1) is solved, whereas Υ represents the lumped-compartment described by a system of ODEs.

Our purpose is to deal with a 3D-0D coupling scheme, however the same procedure can be applied to a 2D-0D coupling, thus we consider a domain $\Omega \in \mathbb{R}^n$ with $n = 2, 3$ and a circuit Υ which unknowns are defined by the state variable vector \underline{y} . The two compartments are connected by a *buffer*, characterized by a linear resistor R_b and a linear capacitor C_b as illustrated by Fig. 7.2. In the domain Ω we solve the fluid part of the model introduced in Ch. 5 (Eqs. 5.13):

$$\frac{1}{M} \frac{\partial p}{\partial t} + \nabla \cdot \underline{j} = f \quad \text{in } \Omega \times (0, T) \quad (7.1a)$$

$$\underline{j} + \underline{\mathcal{K}} \nabla p = \underline{0} \quad \text{in } \Omega \times (0, T) \quad (7.1b)$$

where p is the fluid pressure, \underline{j} is the discharge velocity, f represents the volumetric source term, $\underline{\mathcal{K}}$ is the permeability tensor, and M is the Biot modulus. The system of PDEs (7.1) is endowed with the following boundary conditions

$$\underline{j} \cdot \underline{n} = g_N \quad \text{on } \Gamma_N \times (0, T) \quad (7.2)$$

$$p = g_D \quad \text{on } \Gamma_D \times (0, T) \quad (7.3)$$

and interface conditions with the circuit Υ

$$\int_{\Gamma_I} \underline{u} \cdot \underline{n} = Q_I \text{ on } \Gamma_I \times (0, T) \quad (7.4a)$$

$$p|_{\Gamma_I} = P_I \text{ on } \Gamma_I \times (0, T) \quad (7.4b)$$

$$Q_I(t) = \frac{P_I - \Pi_1}{R_b} \text{ on } \Gamma_I \times (0, T) \quad (7.4c)$$

where $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_I$ with Γ_I being interface between Ω and the buffer, \underline{n} is the outward normal vector, g_N and g_D are the Neumann and Dirichlet boundary condition terms respectively, Q_I is the flux flowing from the domain Ω into the circuit Υ , P_I is the pressure at the interface Γ_I and Π_1 is the unknown pressure that is shared by the buffer and the circuit state variable vector \underline{y} ($\Pi_1 \subseteq \underline{y}$).

The circuit Υ is uniquely described by the following system of ODEs:

$$\frac{d\underline{y}}{dt} + \underline{A}(\underline{y}, t)\underline{y} = \underline{r}(\underline{y}, t) \quad (7.5)$$

with the initial conditions

$$\underline{y}(t = 0) = \underline{y}_0.$$

If d is the number of unknowns of the circuit, *i.e.* the number of ODEs, \underline{y} is a d -dimensional vector representing the state variables of the circuit Υ , \underline{A} is a $d \times d$ tensor including the topology and physics of the connections among the circuit nodes, and

$$\underline{r}(\underline{y}, t) = \underline{s}(\underline{y}, t) + \underline{b}(\underline{y}, t) \quad (7.6)$$

is composed by sources and sinks within the circuit, including generators of current and voltage - denoted \underline{s} - and the contribution due to the coupling with the PDE region Ω - denoted \underline{b} .

State variable Coupling dimension	Units of state variable		Units of differential equation	
	$\Omega \in \mathbb{R}^2$	$\Omega \in \mathbb{R}^3$	$\Omega \in \mathbb{R}^2$	$\Omega \in \mathbb{R}^3$
pressure	$\text{Kg m}^{-1} \text{s}^{-2}$		$\text{Kg m}^{-1} \text{s}^{-2}$	
pressure difference	$\text{Kg m}^{-1} \text{s}^{-2}$		$\text{Kg m}^{-1} \text{s}^{-2}$	
volume	m^2	m^3	$\text{m}^2 \text{s}^{-1}$	$\text{m}^3 \text{s}^{-1}$
flow rate	$\text{m}^2 \text{s}^{-1}$	$\text{m}^3 \text{s}^{-1}$	$\text{m}^2 \text{s}^{-2}$	$\text{m}^3 \text{s}^{-2}$

Table 7.1 – Physical units of state variables for the lumped-parameter model and their corresponding differential equation, in the case of coupling with 2D or 3D computational domain and PDE model.

Remark 4. *In this work we focus on lumped-parameter models involving resistive and capacitive elements, however the theory can be easily extended to include also inductors. This fact leads to choose different electric state variables \underline{y} such as potential, voltage, charge or currents, which, exploiting the fluid analogy, correspond to pressure, volume or volumetric flow rate. Thus, system (7.5) presents different physical units for state variables and equations depending on whether the coupling is with a 2D or 3D PDE domain (see Tab. 7.1).*

Before going through a deeper analysis on this system, let introduce some other notations: for $n = 2$ or 3 , given the vector valued functions $\underline{v}, \underline{w}: (0, T) \rightarrow \mathbb{R}^n$, the vector field $\underline{\varphi}: \Omega \times$

$(0, T) \rightarrow \mathbb{R}^n$ and the tensor field $\underline{\underline{C}}: \Omega \times (0, T) \rightarrow \mathbb{R}^{n \times n}$, we have:

$$\|\underline{v}\|^2 = \sum_{i=1}^n v_i^2 \quad \|\underline{\varphi}\|_{L^2(\Omega)}^2 = \int_{\Omega} \underline{\varphi} \cdot \underline{\varphi} \quad \|\underline{\underline{C}}\|_{L^2(\Omega)}^2 = \int_{\Omega} \underline{\underline{C}} : \underline{\underline{C}} \quad (7.7)$$

where the spaces \mathbb{R}^d and $\mathbb{R}^{d \times d}$ are endowed with the usual Euclidean inner products and where - for sake of clarity - the time dependence is omitted. We denote with the symbol $\langle \cdot, \cdot \rangle$ the inner product and we recall that, given $\underline{\underline{C}} \in \mathbb{R}^{n \times n}$ symmetric and positive definite, the norm $\|\underline{v}\|_{\underline{\underline{C}}} = \sqrt{\langle \underline{v}, \underline{\underline{C}} \underline{v} \rangle}$.

2 Energy of the system

In order to derive the energy of the system, we compute the variational formulation for the PDEs described by (7.1):

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} q + \int_{\Omega} \nabla \cdot \underline{j} q = \int_{\Omega} f q \quad \forall q \in L^2(\Omega);$$

using $\underline{j} = -\underline{\underline{K}} \nabla p$, we obtain

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} q + \int_{\Omega} \nabla \cdot (-\underline{\underline{K}} \nabla p) q = \int_{\Omega} f q \quad \forall q \in L^2(\Omega).$$

Introducing the hypothesis of isotropy $\underline{\underline{K}} = k \underline{I}$ - with $k \geq 0$ for the definition of permeability coefficient - and using Green's formula we infer that

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} q + k \int_{\Omega} \nabla p \cdot \nabla q - k \int_{\partial\Omega} \nabla p q \cdot \underline{n} d\sigma = \int_{\Omega} f q \quad \forall q \in L^2(\Omega)$$

To compute the energy associated to the PDE system, we first choose for the test function $q = p \in H_{0,\Gamma_D}^1(\Omega)$ to obtain the following equation:

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} p + k \int_{\Omega} |\nabla p|^2 - k \int_{\Gamma_N \cup \Gamma_I} \nabla p p \cdot \underline{n} d\sigma = \int_{\Omega} f p.$$

Then, by substituting the BC (7.2) and (7.3), and by using the assumptions on Γ_I that Q_I and P_I are constant in space (Eq. (4.8)), we have

$$\frac{d}{dt} \underbrace{\frac{1}{2M} \|p\|_{L^2(\Omega)}^2}_{\mathcal{E}_{pde}} + \underbrace{k \|\nabla p\|_{L^2(\Omega)}^2}_{\mathcal{D}_{pde}} = \underbrace{\int_{\Omega} f p - \int_{\Gamma_N} g_N p d\sigma - P_I(t) Q_I(t)}_{\mathcal{F}_{pde}} \quad (7.8a)$$

or, equivalently,

$$\frac{d}{dt} \mathcal{E}_{pde} + \mathcal{D}_{pde} = \mathcal{F}_{pde} - P_I(t) Q_I(t) \quad (7.8b)$$

In particular, we have that

- \mathcal{E}_{pde} represents energy in the PDE domain;

- \mathcal{D}_{pde} represents the dissipation due to permeability in the PDE domain;
- \mathcal{F}_{pde} represents the forcing term in the PDE domain;
- $P_I(t) Q_I(t)$ represents the contribution due to the PDE-ODE connection.

Let us now derive a similar energy estimate for the circuit Υ . Since the equations might not be homogeneous in terms of physical units, see Remark 4, we perform the inner product between Eq. (7.5) and the vector valued function $\underline{\underline{U}} \underline{y}$ to have for each of the resulting scalar equations the physical dimensions of a rate of change of energy, namely $\text{Kg m}^2 \text{s}^{-3}$ in case of 3D domain, or that of a rate of change of energy per unit length, namely Kg m s^{-3} in case of 2D domain. The tensor $\underline{\underline{U}}$ is diagonal and its entries $\underline{\underline{U}}_{jj}$ with $j = 1, \dots, d$ depend on the particular choice for the corresponding state variable (see Tab. 7.2); specifically for $j = 1, \dots, d$:

- if y_j is a pressure or a pressure difference, then $\underline{\underline{U}}_{jj}$ is a capacitance;
- if y_j is a volume, then $\underline{\underline{U}}_{jj}$ is an inverse of a capacitance;
- if y_j is a flow rate, then $\underline{\underline{U}}_{jj}$ is an inductance.

y_j	$\underline{\underline{U}}_{jj}$	physical units	
		$\Omega \in \mathbb{R}^2$	$\Omega \in \mathbb{R}^3$
pressure	capacitance	$\text{kg}^{-1} \text{m}^3 \text{s}^2$	$\text{kg}^{-1} \text{m}^4 \text{s}^2$
pressure difference	capacitance	$\text{kg}^{-1} \text{m}^3 \text{s}^2$	$\text{kg}^{-1} \text{m}^4 \text{s}^2$
volume	inverse of a capacitance	$\text{kg m}^{-3} \text{s}^{-2}$	$\text{kg m}^{-4} \text{s}^{-2}$
flow rate	inductance	kg m^{-3}	kg m^{-4}

Table 7.2 – Summary of appropriate choices for the diagonal entries U_{jj} of the tensor $\underline{\underline{U}}$ depending on the corresponding state variable y_j . Thus, the physical units for U_{jj} are listed in case of coupling with a 2D or a 3D domain, respectively.

The specific choice for capacitances or inductances appearing in U_{jj} is determined by the corresponding circuit element pertaining to y_j . Thus, performing the scalar product between Eq. (7.5) and $\underline{\underline{U}} \underline{y}$ we obtain:

$$\underbrace{\left\langle \frac{dy}{dt}, \underline{\underline{U}} \underline{y} \right\rangle}_{\text{I}} + \underbrace{\left\langle \underline{\underline{A}} \underline{y}, \underline{\underline{U}} \underline{y} \right\rangle}_{\text{II}} = \underbrace{\left\langle r, \underline{\underline{U}} \underline{y} \right\rangle}_{\text{III}}. \quad (7.9)$$

We consider the following equality that exploits the fact that $\underline{\underline{U}}$ is diagonal

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \langle \underline{y}, \underline{\underline{U}} \underline{y} \rangle &= \frac{1}{2} \left\langle \frac{dy}{dt}, \underline{\underline{U}} \underline{y} \right\rangle + \frac{1}{2} \left\langle \underline{y}, \frac{d}{dt} (\underline{\underline{U}} \underline{y}) \right\rangle \\ &= \frac{1}{2} \left\langle \frac{dy}{dt}, \underline{\underline{U}} \underline{y} \right\rangle + \frac{1}{2} \left\langle \underline{y}, \frac{d\underline{\underline{U}}}{dt} \underline{y} \right\rangle + \frac{1}{2} \left\langle \underline{y}, \underline{\underline{U}} \frac{dy}{dt} \right\rangle \\ &= \left\langle \frac{dy}{dt}, \underline{\underline{U}} \underline{y} \right\rangle + \frac{1}{2} \left\langle \underline{y}, \frac{d\underline{\underline{U}}}{dt} \underline{y} \right\rangle. \end{aligned}$$

Substituting this result in I of (7.9), we have:

$$\left\langle \frac{dy}{dt}, \underline{\underline{U}} \underline{y} \right\rangle = \frac{1}{2} \frac{d}{dt} \langle \underline{y}, \underline{\underline{U}} \underline{y} \rangle - \underbrace{\frac{1}{2} \left\langle \underline{y}, \frac{d\underline{\underline{U}}}{dt} \underline{y} \right\rangle}_{\text{IV}}. \quad (7.10)$$

Therefore we introduce the matrix $\underline{\underline{B}}$ summing II of (7.9) and IV of (7.10):

$$\langle \underline{\underline{A}} \underline{y}, \underline{\underline{U}} \underline{y} \rangle - \frac{1}{2} \langle \underline{y}, \frac{d\underline{\underline{U}}}{dt} \underline{y} \rangle = \langle \underline{y}, \underline{\underline{U}} \underline{\underline{A}} \underline{y} \rangle - \frac{1}{2} \langle \underline{y}, \frac{d\underline{\underline{U}}}{dt} \underline{y} \rangle = \langle \underline{y}, \underline{\underline{B}} \underline{y} \rangle$$

where

$$\underline{\underline{B}} = \underline{\underline{U}} \underline{\underline{A}} - \frac{1}{2} \frac{d}{dt} \underline{\underline{U}}. \quad (7.11)$$

Moreover we rewrite III of (7.9) as

$$\langle \underline{r}, \underline{\underline{U}} \underline{y} \rangle = \langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle + \langle \underline{s}, \underline{\underline{U}} \underline{y} \rangle.$$

Finally, we obtain

$$\frac{d}{dt} \underbrace{\frac{1}{2} \|\underline{y}\|_{\underline{\underline{U}}}}_{\mathcal{E}_{ode}} + \underbrace{\langle \underline{y}, \underline{\underline{B}} \underline{y} \rangle}_{\mathcal{U}_{ode}} = \underbrace{\langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle}_{\mathcal{F}_{ode}} + \underbrace{\langle \underline{s}, \underline{\underline{U}} \underline{y} \rangle}_{\mathcal{F}_{ode}} \quad (7.12a)$$

$$\frac{d}{dt} \mathcal{E}_{ode} + \mathcal{U}_{ode} = \mathcal{F}_{ode} + \langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle \quad (7.12b)$$

where

- \mathcal{E}_{ode} represents the energy characterizing the 0D circuit;
- \mathcal{U}_{ode} represents all the contributions from resistors and capacitors in the 0D circuit;
- \mathcal{F}_{ode} represents the forcing on the 0D circuit system due to current or voltage generators;
- $\langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle$ represents the contribution due to the PDE-ODE connection.

If we combine (7.8) and (7.12), we obtain the following energy balance equation:

$$\frac{d}{dt} (\mathcal{E}_{pde} + \mathcal{E}_{ode}) + \mathcal{D}_{pde} + \mathcal{U}_{ode} = \mathcal{F}_{pde} + \mathcal{F}_{ode} + \underbrace{\langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle - P_I(t) Q_I(t)}_{\mathcal{G}} \quad (7.13)$$

Due to the resistive-capacitive coupling connection we have chosen to interface the domain Ω with the lumped-parameter compartment Υ (see Fig. 7.2), we have that $\langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle = \Pi_1 Q_I$. Considering Eq. (7.4c), we obtain

$$\begin{aligned} \mathcal{G} &= \langle \underline{b}, \underline{\underline{U}} \underline{y} \rangle - P_I Q_I \\ &= \Pi_1 Q_I - P_I Q_I \\ &= -R_b Q_I^2 \\ &= -\mathcal{D}_{RC} \end{aligned}$$

Finally, we deduce that

$$\frac{d}{dt} (\mathcal{E}_{pde} + \mathcal{E}_{ode}) + \mathcal{D}_{pde} + \mathcal{U}_{ode} + \mathcal{D}_{RC} = \mathcal{F}_{pde} + \mathcal{F}_{ode} \quad (7.14)$$

Considering that $\mathcal{D}_{pde} + \mathcal{D}_{RC} \geq 0 \quad \forall t \geq 0$, if \mathcal{F}_{pde} and \mathcal{F}_{ode} are vanishing and the circuit properties are such that $\mathcal{U}_{ode} \geq 0 \quad \forall t \geq 0$, then we have $\frac{d}{dt} (\mathcal{E}_{pde} + \mathcal{E}_{ode}) \leq 0 \quad \forall t \geq 0$, which leads to the inequality

$$\mathcal{E}_{pde}(t) + \mathcal{E}_{ode}(t) \leq \mathcal{E}_{pde}(0) + \mathcal{E}_{ode}(0) \quad \forall t \in (0, T) \quad (7.15)$$

Eq. 7.15 indicates that the physical energy of the system in the absence of external forces decays in time. Our goal is to devise a numerical method that preserves this property at the discrete level, thereby ultimately yielding **unconditional stability** with respect to the time discretization. This property is extensively discussed in Sec. 5.

3 Dimensional analysis

In this section we complete a dimensional analysis in the case of the coupling of a 3D PDE model with a 0D circuit. Using the information provided by Tab. 7.1 it is straightforward to achieve a similar results for a 2D PDE model.

We compute the dimension units for the main energy entries of the 3D PDE model:

- energy

$$\begin{aligned} \mathcal{E}_{pde} &= \left[\frac{1}{M} \right] \left[\int_{\Omega} \frac{d}{dt} \right] [p]^2 &&= \frac{\text{ms}^2}{\text{Kg}} \frac{\text{m}^3}{\text{s}} \left(\frac{\text{N}}{\text{m}^2} \right)^2 \\ &= \frac{\text{m}^4 \text{s}}{\text{Kg}} \frac{\text{m}^2 \text{Kg}^2}{\text{s}^4 \text{m}^4} &&= \boxed{\frac{\text{m}^2 \text{Kg}}{\text{s}^3}} \end{aligned}$$

- dissipation

$$\begin{aligned} \mathcal{D}_{pde} &= [\underline{\mathcal{K}}] \left[\int_{\Omega} \right] [\nabla p]^2 &&= \frac{\text{m}^3 \text{s}}{\text{Kg}} \text{m}^3 \left(\frac{1}{\text{m}} \frac{\text{N}}{\text{m}^2} \right)^2 \\ &= \frac{\text{m}^6 \text{s}}{\text{Kg}} \frac{1}{\text{m}^6} \frac{\text{m}^2 \text{Kg}^2}{\text{s}^4} &&= \boxed{\frac{\text{m}^2 \text{Kg}}{\text{s}^3}} \end{aligned}$$

As an example we perform also the dimensional analysis of the energy of the 0D circuit for different types of state variables (see Tabs. 7.1 and 7.2 for the units):

- pressure or pressure difference

$$\begin{aligned} \mathcal{E}_{ode} &= \left[\frac{d}{dt} \right] [U_j] [y_j]^2 &&= \frac{1}{\text{s}} [C] \frac{\text{Kg}^2}{\text{m}^2 \text{s}^4} \\ &= \frac{\text{Kg}^2}{\text{m}^2 \text{s}^5} \frac{\text{m}^4 \text{s}^2}{\text{Kg}} &&= \boxed{\frac{\text{m}^2 \text{Kg}}{\text{s}^3}} \end{aligned}$$

- volume

$$\mathcal{E}_{ode} = \left[\frac{d}{dt} \right] [U_j] [y_j]^2 = \frac{1}{\text{s}} \left[\frac{1}{C} \right] \text{m}^6$$

$$= \frac{\text{m}^6}{\text{s}} \frac{\text{Kg}}{\text{m}^4 \text{s}^2} = \boxed{\frac{\text{m}^2 \text{Kg}}{\text{s}^3}}$$

- flow rate

$$\begin{aligned} \mathcal{E}_{ode} &= \left[\frac{d}{dt} \right] [U_j] [y_j]^2 = \frac{1}{\text{s}} [L] \frac{\text{m}^6}{\text{s}^2} \\ &= \frac{\text{m}^6}{\text{s}^3} \frac{\text{Kg}}{\text{m}^4} = \boxed{\frac{\text{m}^2 \text{Kg}}{\text{s}^3}} \end{aligned}$$

Analogously, this procedure can be performed on all the other energy entries both for the PDE and the ODE part using any type of state variable, achieving - for the case $\Omega \in \mathbb{R}^3$ - the units $\frac{\text{m}^2 \text{Kg}}{\text{s}^3}$.

4 Operator splitting technique

The PDE-ODE coupled system presented in Ch. 4 presents several nonlinearities, for instance the interaction between IOP/RLTp and the blood pressure within the CRV (Eq. (3.8)). Moreover due to the multiscale character of the problem, the coupling between the lamina cribrosa and the blood circulation in the back of the eye is very challenging. To overcome these difficulties, numerous coupling scheme have been already developed in various contexts. For instance in [33, 60, 192, 80, 260, 70] the authors propose robust and stable methods based on implicit or fully coupled algorithms, which are, however, quite demanding in terms of computational time. Other approaches are based on semi-implicit schemes [69, 189], monolithic-like schemes [170] or explicit coupling scheme with the use of a penalty term [26]. Thus, the idea of splitting our problem is not new in the literature. The novelty of the technique we are presenting is how to implement the splitting for a Darcy-ODEs system with the goal of preserving the energy identity at the continuous level.

Let now consider the 3D problem described by Eq. (7.1) coupled with the 0D model defined by Eq. (7.5). We approximate it in time by using the Lie's scheme [84, 85]. The Lie's scheme can be summarized as follows. Consider the following initial value problem:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + \mathcal{A}(\phi) &= 0 \quad \text{in } (0, T), \\ \phi(0) &= \phi_0, \end{aligned} \tag{7.16}$$

where \mathcal{A} is a nonlinear operator from a Hilbert space into itself. Suppose that operator \mathcal{A} has a non-trivial decomposition

$$\mathcal{A} = \sum_{i=1}^I \mathcal{A}_i \tag{7.17}$$

Then, the solution of the initial value problem 7.16 can be approximated *via* the following scheme:

Let $\Delta t > 0$ be a time-discretization step. Denote $t^n = n \Delta t$ and let ϕ^n be an approximation of $\phi(t^n)$. Set $\phi^0 = \phi_0$. Then, for $n \geq 0$ compute ϕ^{n+1} by solving

$$\begin{aligned} \frac{\partial \phi_i}{\partial t} + \mathcal{A}_i(\phi_i) &= 0 \quad \text{in } (t^n, t^{n+1}), \\ \phi(t^n) &= \phi^{n+(i-1)/I}, \end{aligned} \quad (7.18)$$

and then set $\phi^{n+i/I} = \phi_i(t^{n+1})$, for $i = 1, \dots, I$. This method is first-order accurate. More precisely, if (7.16) is defined on a finite-dimensional space and if the operators \mathcal{A}_i are smooth enough, then

$$\|\phi(t^n) - \phi^n\| = O(\Delta t).$$

Our 3D-0D problem can be thought as the analogous to problem (7.16), where ϕ is the array of unknowns p , j and \underline{y} , while \mathcal{A} is a multivalued nonlinear differential operator. There is not a unique way to decompose \mathcal{A} (Eq. (7.17)), and different choices may lead to the solution of different sub-problems (Eq. (7.18)).

Our strategy is to solve separately the following problems:

1. the 3D Biot-Darcy problem (Eq. (7.1));
2. the nonlinear 0D circuit described by a system of ODEs (Eq. (7.5)).

Note that our goal is to separate the nonlinearities of the ODE system from the PDE system and to split the 3D domain from the 0D circuit. In this way, we obtain two sub-systems in which we can employ different spatial and temporal discretization and use adapted numerical methods. Thus, in order to preserve the energy at the continuous level, we solve in a first step the PDE system accompanied by the interface conditions and the contributions due to the 3D-0D connection, while in a second step we solve the remaining ODE system.

Details of the splitting are presented next.

5 Two-step splitting

Let $t^n = n\Delta t$ with Δt the time step. We introduce the discrete notation for the pressure in the 2D/3D domain Ω and the vector of unknowns in the circuit Υ :

$$p^n = p(t^n) \qquad \underline{y}^n = \underline{y}(t^n).$$

Considering the energy estimate presented in Sec. 2 at the continuous level, we infer that to achieve unconditional stability in time, we can adopt the following two-step approach:

given p^n, \underline{y}^n for $n \geq 0$, solve

Step 1.

$$\begin{aligned} \frac{1}{M} \frac{\partial p}{\partial t} + \nabla \cdot \underline{j} &= f & \text{in } \Omega \times (t^n, t^{n+1}) \\ \underline{K} \nabla p + \underline{j} &= \underline{0} & \text{in } \Omega \times (t^n, t^{n+1}) \\ \frac{d\underline{y}}{dt} &= \underline{b}(\underline{y}, t) & \text{in } (t^n, t^{n+1}) \end{aligned}$$

+ interface, boundary and initial conditions

$$p(t^n) = p^n \quad \underline{y}(t^n) = \underline{y}^n$$

and then set

$$\begin{aligned} p^{n+\frac{1}{2}} &= p(t^{n+\frac{1}{2}}) \\ \underline{y}^{n+\frac{1}{2}} &= \underline{y}(t^{n+\frac{1}{2}}) \\ \underline{j}^{n+\frac{1}{2}} &= \underline{j}(t^{n+\frac{1}{2}}) \end{aligned}$$

Step 2.

$$\frac{\partial p}{\partial t} = 0 \quad \text{in } \Omega \times (t^n, t^{n+1})$$

$$\frac{d\underline{y}}{dt} + \underline{A} \underline{y} = \underline{s}(t) \quad \text{in } (t^n, t^{n+1})$$

+ initial conditions

$$p(t^n) = p^{n+\frac{1}{2}} \quad \underline{y}(t^n) = \underline{y}^{n+\frac{1}{2}}$$

and then set

$$\begin{aligned} p^{n+1} &= p(t^{n+1}) \quad (= p(t^{n+\frac{1}{2}})) \\ \underline{y}^{n+1} &= \underline{y}(t^{n+1}) \\ \underline{j}^{n+1} &= \underline{j}(t^{n+1}) \quad (= \underline{j}(t^{n+\frac{1}{2}})) \end{aligned}$$

In particular we emphasize the splitting we applied to the forcing term for the circuit system in Eq. (7.6), *i.e.* we have decoupled the contributions due to the interface between Ω and Υ ($\underline{b}(\underline{y}, t)$) and contributions that derive from the sources and sinks within the circuit ($\underline{s}(\underline{y}, t)$).

We compute now the energy identity for the proposed two-step algorithm. First, the variational formulation for the first equation in Step 1:

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} q + \int_{\Omega} \nabla \cdot \underline{j} q = \int_{\Omega} f q \quad \forall q \in L^2(\Omega) \quad t \in (t^n, t^{n+1})$$

using $\underline{j} = -k\underline{I} \nabla p$ (hypothesis of isotropy) and the Green's formula we obtain

$$\int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} q + k \int_{\Omega} \nabla p \cdot \nabla q - k \int_{\partial\Omega} \nabla p q \cdot \underline{n} d\sigma = \int_{\Omega} f q \quad \forall q \in L^2(\Omega) \quad t \in (t^n, t^{n+1}).$$

To compute the energy associated to Step 1, we first choose for the test function $q = p \in H_{0,\Gamma_D}^1(\Omega)$ for the first equation and we perform the scalar product between the third equation of Step 1 and $\underline{U} \underline{y}$. Thus, we obtain:

$$\begin{cases} \int_{\Omega} \frac{1}{M} \frac{\partial p}{\partial t} p + k \int_{\Omega} |\nabla p|^2 - k \int_{\Gamma_N \cup \Gamma_I} \nabla p p \cdot \underline{n} d\sigma = \int_{\Omega} f p & t \in (t^n, t^{n+1}) \\ \langle \frac{d\underline{y}}{dt}, \underline{U} \underline{y} \rangle = \langle \underline{b}, \underline{U} \underline{y} \rangle & t \in (t^n, t^{n+1}) \end{cases}$$

Then, we use Eqs. (7.2), (7.3) and (4.8) on the first equation, and we sum it to the second equation, we have for $t \in (t^n, t^{n+1})$

$$\frac{d}{dt} \underbrace{\left(\frac{1}{2} \frac{1}{M} \|p\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\underline{y}\|_{\underline{U}} \right)}_{\mathcal{E}_1} + \underbrace{k \|\nabla p\|_{L^2(\Omega)}^2}_{\mathcal{D}_1} = \underbrace{\int_{\Omega} f p - \int_{\Gamma_N} g_N p d\sigma}_{\mathcal{F}_1} + \underbrace{\langle \underline{b}, \underline{U} \underline{y} \rangle - P_I(t) Q_I(t)}_{-\mathcal{G}_1} \quad (7.19a)$$

or, equivalently,

$$\frac{d}{dt} \mathcal{E}_1 + \mathcal{D}_1 + \mathcal{G}_1 = \mathcal{F}_1 \quad t \in (t^n, t^{n+1}). \quad (7.19b)$$

where

- \mathcal{E}_1 represents energy in Step 1;
- \mathcal{D}_1 represents the dissipation in the PDE domain;
- \mathcal{G}_1 represents the contribution due to the PDE-ODE connection;
- \mathcal{F}_1 represents the forcing term in the PDE domain.

Considering that $\mathcal{D}_1 \geq 0 \quad \forall t \in (t^n, t^{n+1})$ and $\mathcal{G}_1 = -\langle \underline{b}, \underline{U} \underline{y} \rangle + P_I(t) Q_I(t) = R_b Q_I^2 \geq 0 \quad \forall t \in (t^n, t^{n+1})$, if \mathcal{F}_1 is vanishing, then we have $\frac{d}{dt} \mathcal{E}_1 \leq 0 \quad \forall t \in (t^n, t^{n+1})$ which leads to the inequality

$$\mathcal{E}_1(t^{n+1}) \leq \mathcal{E}_1(t^n) \quad \forall n \geq 0 \quad (7.20)$$

Eq. 7.20 indicates that the physical energy of Step 1 in the absence of external force decays in time.

Similarly, for Step 2 we obtain:

$$\begin{cases} \int_{\Omega} \frac{\partial p}{\partial t} p = 0 & t \in (t^n, t^{n+1}) \\ \left\langle \frac{d\underline{y}}{dt}, \underline{U} \underline{y} \right\rangle + \langle \underline{A} \underline{y}, \underline{U} \underline{y} \rangle = \langle \underline{s}, \underline{U} \underline{y} \rangle & t \in (t^n, t^{n+1}) \end{cases}$$

Then, we use the same steps as for the continuous case, and we sum the two equations, obtaining:

$$\frac{d}{dt} \underbrace{\left(\frac{1}{2} \|p\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\underline{y}\|_{\underline{U}} \right)}_{\mathcal{E}_2} + \underbrace{\langle \underline{y}, \underline{B} \underline{y} \rangle}_{\mathcal{U}_2} = \underbrace{\langle \underline{s}, \underline{U} \underline{y} \rangle}_{\mathcal{F}_2} \quad (7.21a)$$

$$(7.21b)$$

or, equivalently,

$$\frac{d}{dt} \mathcal{E}_2 + \mathcal{U}_2 = \mathcal{F}_2 \quad t \in (t^n, t^{n+1}). \quad (7.21c)$$

where

- \mathcal{E}_2 represents the energy characterizing Step 2;
- \mathcal{U}_2 represents all the contributions from resistors and capacitors in the 0D circuit;
- \mathcal{F}_2 represents the forcing on the 0D circuit system due to current or voltage generators.

If \mathcal{F}_2 is vanishing and the circuit properties are such that $\mathcal{U}_2 \geq 0 \quad \forall t \in (t^n, t^{n+1})$, then we have $\frac{d}{dt} \mathcal{E}_2 \leq 0 \quad \forall t \in (t^n, t^{n+1})$ which leads to the inequality

$$\mathcal{E}_2(t^{n+1}) \leq \mathcal{E}_2(t^n) \quad \forall n \geq 0 \quad (7.22)$$

Eq. 7.22 indicates that the physical energy of Step 2 in the absence of external force decays in time.

Theorem 2 (Semi-discrete energy). *Consider the proposed two-step algorithm, i.e. the semi-discretization in time of the 3D problem described by Eq. (7.1) coupled with the 0D model defined by Eq. (7.5), and assume that:*

- (i) *the circuit properties are such that the tensor $\underline{\underline{A}}$ is constant and $\underline{\underline{B}}$ is positive definite;*
- (ii) *there are no forcing terms, namely $\mathcal{F}_1(t^n) = \mathcal{F}_2(t^n) = 0 \forall n \geq 0$, where \mathcal{F}_1 represents the forcing term in Step 1 and \mathcal{F}_2 the forcing term in Step 2.*

Under these assumptions, each step of the proposed algorithm preserves the energy at each interval $(t^n, t^{n+1}) \forall n \geq 0$ and is unconditionally stable.

Remark 5. *As a results of this theorem, the proposed strategy is very modular, allowing notable flexibility in the choice of solution methods for each sub-problem resulting from the splitting. The complexity coincides with the one of loosely coupled schemes, due to the fact that no iterations between the two main steps are necessary to achieve stability.*

As a result of this theoretical investigation we obtain, in a simplified case where we employ implicit Euler for the temporal discretization, the following theorem:

Theorem 3 (Stability). *Consider the 3D problem described by Eq. (7.1) coupled with the 0D model defined by Eq. (7.5) and assume that:*

- (i) *the circuit properties are such that the tensor $\underline{\underline{A}}$ is constant and $\underline{\underline{B}}$ is positive definite;*
- (ii) *there are no forcing terms, namely $\mathcal{F}_{pde}(t) = \mathcal{F}_{ode}(t) = 0 \forall t \geq 0$, where \mathcal{F}_{pde} represents the forcing on the system due to body forces and external pressures acting on the PDE domain and \mathcal{F}_{ode} the forcing on the system due to generators of current and voltage within the lumped circuit, respectively;*
- (iii) *the PDE-ODE connection is of resistive-capacitive type, as displayed in Fig. 7.2;*
- (iv) *implicit Euler has been employed for the discretization in time.*

Under these assumptions, the algorithm given by Steps 1 and 2 is unconditionally stable.

Proof. Let $\Delta t = t^{n+1} - t^n$ and let us begin by considering Step 1. Under assumptions (i)-(ii), using an implicit Euler scheme for the time discretization and following a similar procedure to that detailed in Sec. 2, we obtain the following energy identity at the time discrete level for Step 1:

$$\frac{1}{\Delta t} \mathcal{E}_1^{n+\frac{1}{2}} + \mathcal{D}_{pde,1}^{n+\frac{1}{2}} + \mathcal{D}_{RC,1}^{n+\frac{1}{2}} = \frac{1}{\Delta t} \left(\frac{1}{M} \int_{\Omega} p^n p^{n+\frac{1}{2}} + \langle \underline{y}^n, \underline{\underline{U}} \underline{y}^{n+\frac{1}{2}} \rangle \right) \quad (7.23)$$

where $\mathcal{E}_1^{n+\frac{1}{2}} = \mathcal{E}_{pde,1}^{n+\frac{1}{2}} + \mathcal{E}_{ode,1}^{n+\frac{1}{2}}$ and

$$\begin{aligned} \mathcal{E}_{pde,1}^{n+\frac{1}{2}} &= \frac{1}{M} \left\| p^{n+\frac{1}{2}} \right\|_{L^2(\Omega)}^2 & \mathcal{E}_{ode,1}^{n+\frac{1}{2}} &= \left\| \underline{y}^{n+\frac{1}{2}} \right\|_{\underline{\underline{U}}}^2 \\ \mathcal{D}_{pde,1}^{n+\frac{1}{2}} &= k \left\| \nabla p^{n+\frac{1}{2}} \right\|_{\mathbf{L}^2(\Omega)}^2 & \mathcal{D}_{RC,1}^{n+\frac{1}{2}} &= R_b \left(Q_I^{n+\frac{1}{2}} \right)^2 \end{aligned}$$

Using Young's inequality on the first term on the right hand side of (7.23), yields

$$\frac{1}{M} \int_{\Omega} p^n p^{n+\frac{1}{2}} \leq \frac{1}{2M} \left(\|p^n\|_{L^2(\Omega)}^2 + \|p^{n+\frac{1}{2}}\|_{L^2(\Omega)}^2 \right) = \frac{1}{2} \mathcal{E}_{pde,1}^n + \frac{1}{2} \mathcal{E}_{pde,1}^{n+\frac{1}{2}} \quad (7.24)$$

To estimate the second term on the right hand side of (7.23), first we use Cauchy-Schwarz inequality, then again Young's inequality to obtain

$$\begin{aligned} \langle \underline{y}^n, \underline{U} \underline{y}^{n+\frac{1}{2}} \rangle &\leq \langle \underline{y}^n, \underline{U} \underline{y}^n \rangle \langle \underline{y}^{n+\frac{1}{2}}, \underline{U} \underline{y}^{n+\frac{1}{2}} \rangle \\ &\leq \frac{1}{2} \left(\langle \underline{y}^n, \underline{U} \underline{y}^n \rangle + \langle \underline{y}^{n+\frac{1}{2}}, \underline{U} \underline{y}^{n+\frac{1}{2}} \rangle \right) \\ &= \frac{1}{2} \mathcal{E}_{ode,1}^n + \frac{1}{2} \mathcal{E}_{ode,1}^{n+\frac{1}{2}} \end{aligned} \quad (7.25)$$

These operations are possible since the matrix \underline{U} is symmetric and positive definite. Combining (7.23), (7.24) and (7.25), we retrieve

$$\begin{aligned} \frac{1}{2\Delta t} \mathcal{E}_1^{n+\frac{1}{2}} + \mathcal{D}_{pde,1}^{n+\frac{1}{2}} + \mathcal{D}_{RC,1}^{n+\frac{1}{2}} &\leq \frac{1}{2\Delta t} \mathcal{E}_1^n \\ \mathcal{E}_1^{n+\frac{1}{2}} - \mathcal{E}_1^n &\leq -2\Delta t \left(\mathcal{D}_{pde,1}^{n+\frac{1}{2}} + \mathcal{D}_{RC,1}^{n+\frac{1}{2}} \right) \\ \mathcal{E}_1^{n+\frac{1}{2}} - \mathcal{E}_1^n &\leq 0 \end{aligned}$$

Finally we conclude that

$$\mathcal{E}_1^{n+\frac{1}{2}} \leq \mathcal{E}_1^n \quad (7.26)$$

Let now consider the energy identity for Step 2. We use implicit Euler scheme for the time discretization and deduce that:

$$\frac{1}{\Delta t} \mathcal{E}_2^{n+1} + \mathcal{U}_{ode,2}^{n+1} = \frac{1}{\Delta t} \left(\frac{1}{M} \int_{\Omega} p^{n+\frac{1}{2}} p^{n+1} + \langle \underline{y}^{n+\frac{1}{2}}, \underline{U} \underline{y}^{n+1} \rangle \right) \quad (7.27)$$

at the time discrete level for Step 2, where $\mathcal{E}_2^{n+1} = \mathcal{E}_{pde,2}^{n+1} + \mathcal{E}_{ode,2}^{n+1}$ and

$$\mathcal{E}_{pde,2}^{n+1} = \frac{1}{M} \|p^{n+1}\|_{L^2(\Omega)}^2 \quad \mathcal{E}_{ode,2}^{n+1} = \left\| \underline{U}^{\frac{1}{2}} \underline{y}^{n+1} \right\|^2 \quad \mathcal{U}_{ode,2}^{n+1} = \left\| \underline{B}^{\frac{1}{2}} \underline{y}^{n+1} \right\|^2$$

To estimate the right hand side of (7.27) we adopt the same procedure used for Step 1 in (7.24) and (7.25), thereby retrieving the following inequality

$$\begin{aligned} \frac{1}{2\Delta t} \mathcal{E}_2^{n+1} + \mathcal{U}_{ode,2}^{n+1} &\leq \frac{1}{2\Delta t} \left(\frac{1}{M} \|p^{n+\frac{1}{2}}\|_{L^2(\Omega)}^2 + \left\| \underline{U}^{\frac{1}{2}} \underline{y}^{n+\frac{1}{2}} \right\|^2 \right) \\ &= \frac{1}{2\Delta t} \mathcal{E}_2^{n+\frac{1}{2}} \end{aligned} \quad (7.28)$$

Since $\mathcal{U}_{ode,2}^{n+1} \geq 0$ from assumption (i), (7.28) leads to

$$\begin{aligned} \mathcal{E}_2^{n+1} - \mathcal{E}_2^{n+\frac{1}{2}} &\leq -2\Delta t \mathcal{U}_{ode,2}^{n+1} \leq 0 \\ \implies \mathcal{E}_2^{n+1} &\leq \mathcal{E}_2^{n+\frac{1}{2}} \end{aligned} \quad (7.29)$$

The fact that initial conditions for Step 2 coincide with the final solution of Step 1 implies that $\mathcal{E}_1^{n+\frac{1}{2}} = \mathcal{E}_2^{n+\frac{1}{2}}$; thus, combining (7.26) and (7.29), we obtain the following inequality

$$\mathcal{E}_2^{n+1} \leq \mathcal{E}_1^{n+\frac{1}{2}} = \mathcal{E}_2^{n+\frac{1}{2}} \leq \mathcal{E}_1^n \quad \text{for } n \geq 1, \quad (7.30)$$

which provides an upper bound for the norm of the solution regardless of the time step size, thereby ensuring **unconditional stability** of the algorithm. \square

Remark 6. *In Step 1 of the suggested algorithm we treat implicitly the contributions from the 3D-0D connection, represented by $\underline{b}(P_I, Q_I, t)$. This choice allows us to evaluate all the quantities in \mathcal{G} at the same time, providing a defined expression for the dissipation term \mathcal{D}_{RC} in the energy estimate even at the discrete level. Thus, we propose a two-step operator splitting scheme that ensures numerical stability with respect to the time discretization without introducing uncontrolled artificial terms in the energy [12, 74].*

Remark 7. *The proposed two-step operator splitting algorithm exhibits a first order convergence in time [84]. Second order convergence could be achieved via strong symmetrization, in combination with time discretization algorithms that are at least second order in time [84].*

 Hybridizable Discontinuous Galerkin (HDG) method

Contents

1	Introductory notation	72
2	Error analysis	74
3	Static condensation	76
3.1	Monolithic solver	77
3.2	Local solvers	78
3.3	Flux operators	78
3.4	Global solver and boundary conditions	79
4	Connectivity of the HDG method with IBC	80
5	Linear elasticity	81

In this chapter we present our contribution to the Hybridizable Discontinuous Galerkin (HDG) method, which is also available in [13]. We adopt this discretization, since one of the biggest issue in biological computation, in particular for small systems, is the high accuracy that all the models require to be able to reproduce physiological behaviors. This type of mixed finite element method is not new in the literature [49], but our contribution in the numerical analysis world is the theorization and implementation of the Integral Boundary Condition (IBC) for HDG.

The advantages of HDG are:

- (i) the equation are enforced element-by-element leading to locally conservative property;
- (ii) optimal convergence properties for both primal and dual variables.

These features induce a huge computational cost, which can be largely mitigated using static condensation described in Sec. 3.

Remark 8. The use of HDG formulation combined with IBC supports the direct solution of the pressure on that boundary without any sub-iteration.

1 Introductory notation

Let consider the following elliptic model elliptic boundary value problem of second order written in Eq. 4.1. In view of the finite element approximation, we partition Ω into the union of d -simplices K , closed straight triangles if $d = 2$ and closed straight tetrahedra if $d = 3$, and we denote by \mathcal{T}_h - called triangulation - the collection of elements K such that $\Omega = \cup_{K \in \mathcal{T}_h} K$. For each $K \in \mathcal{T}_h$, we denote by h_K the diameter of K . We let $h := \max_{K \in \mathcal{T}_h} h_K$ and we consider a family of conforming, regular triangulations of Ω , $\{\mathcal{T}_h\}_{h>0}$ (see [193, Chapter 3]). For each element $K \in \mathcal{T}_h$ we indicate by ∂K the boundary of K and by $\vec{n}_{\partial K}$ the associated outward unit normal vector \underline{n} . The d -dimensional measure of the element K is indicated with $|K|$ while the $(d - 1)$ -dimensional measure of each face - edge if $d = 2$ - of ∂K is denoted by $|F|$. To simplify the exposition we take the convention to use word *face* to refer to F even when $d = 2$.

We let \mathcal{F}_h denote the collection of all the faces of \mathcal{T}_h , whose union forms the *skeleton* of the decomposition \mathcal{T}_h . The set \mathcal{F}_h naturally splits into the subset \mathcal{F}_h^Γ of faces belonging to Γ , boundary of domain Ω , and into the subset of faces belonging to the interior of Ω , denoted by \mathcal{F}_h^0 . Finally, assuming that the decomposition \mathcal{T}_h is such that for all faces F in \mathcal{F}_h with $F \subset \Gamma$ either $F \subset \Gamma_D$, $F \subset \Gamma_N$ or $F \subset \Gamma_I$, the sets of boundary faces can be split into the subsets $\mathcal{F}_h^{\Gamma_D}$, $\mathcal{F}_h^{\Gamma_N}$, and $\mathcal{F}_h^{\Gamma_I}$, which correspond to the boundaries where we impose Dirichlet, Neumann and Integral condition, respectively.

If we denote \underline{q} a vector-valued function, we can define the *jump* of the normal trace of \underline{q} across F as

$$[[\underline{q}]]_F := \underline{q}^{K_1} \cdot \underline{n}_{\partial K_1}|_F + \underline{q}^{K_2} \cdot \underline{n}_{\partial K_2}|_F. \quad (8.1)$$

where K_1 and K_2 are two adjacent elements.

Let us now introduce the finite element spaces

$$V_h = \prod_{K \in \mathcal{T}_h} V(K), \quad W_h = \prod_{K \in \mathcal{T}_h} W(K) \quad (8.2a)$$

$$M_h = \text{span}\langle \phi^* \rangle \oplus \prod_{F \in \mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N}} M(F) \quad (8.2b)$$

with $\phi^* \in L^2(\mathcal{F}_h)$ defined as

$$\phi^*|_F = 1 \quad \forall F \in \mathcal{F}_h^{\Gamma_I}, \quad \phi^*|_F = 0 \quad \forall F \in \mathcal{F}_h \setminus \mathcal{F}_h^{\Gamma_I},$$

and with

$$V(K) = \mathbf{P}_k(K), \quad W(K) = \mathcal{P}_k(K), \quad M(F) = \mathcal{P}_k(F), \quad (8.3)$$

where $\mathcal{P}_k(K)$ (resp. $\mathcal{P}_k(F)$) denote the space of polynomials of degrees less or equal that k on K (resp. F), and where $\mathbf{P}_k(K) = (\mathcal{P}_k(K))^d$. Definitions (8.2) and (8.3) imply that functions belonging to V_h and W_h are, in general, discontinuous across element edges in 2D and elements faces in 3D of \mathcal{T}_h , while functions in M_h are discontinuous across element vertices in 2D and

element edges in 3D of $\mathcal{F}_h \setminus \mathcal{F}_h^{\Gamma_I}$, *single-valued* on each face $F \in \mathcal{F}_h$ of the skeleton of \mathcal{T}_h and constant on Γ_I .

The functions of M_h play the role of “connectors” between adjacent elements that, otherwise, would be irremediably uncoupled in the discretization process. For a mechanical interpretation of the interelement connecting role of the space M_h , see [35] and references cited therein.

Defining the *numerical normal flux* on ∂K

$$\widehat{j}_h^{\partial K} \cdot \underline{n}_{\partial K} = \underline{j}_h^K|_{\partial K} \cdot \underline{n}_{\partial K} + \tau_{\partial K}(p_h^K|_{\partial K} - \widehat{p}_h|_{\partial K}), \quad (8.4)$$

the discrete formulation reads: find $\underline{j}_h \in V_h, p_h \in W_h$ and $\widehat{p}_h \in M_h$ such that $\forall \underline{v}_h \in V_h, \forall w_h \in W_h$ and $\forall \mu_h \in M_h$

$$\sum_{K \in \mathcal{T}_h} (\mathcal{K}^{-1} \underline{j}_h^K, \underline{v}_h^K)_K + (p_h^K, \nabla \cdot \underline{v}_h^K)_K + \langle \widehat{p}_h, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K} = 0, \quad (8.5a)$$

$$\sum_{K \in \mathcal{T}_h} \left[-(\underline{j}_h^K, \nabla w_h^K)_K + \langle \widehat{j}_h^{\partial K} \cdot \underline{n}_{\partial K}, w_h^K \rangle_{\partial K} \right] = \sum_{K \in \mathcal{T}_h} (f, w_h^K)_K, \quad (8.5b)$$

$$\sum_{K \in \mathcal{T}_h} \langle \widehat{j}_h^{\partial K} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K} = \langle g_N, \mu_h \rangle_{\Gamma_N} + I_{target} |\Gamma_I|^{-1} \langle \mu_h, 1 \rangle_{\Gamma_I}, \quad (8.5c)$$

$$\sum_{K \in \mathcal{T}_h} \langle \widehat{p}_h, \mu_h \rangle_{\Gamma_D} = \langle g_D, \mu_h \rangle_{\Gamma_D}. \quad (8.5d)$$

The dependent variables \underline{j}_h and p_h are the approximations of \underline{j} and p in the interior of each element $K \in \mathcal{T}_h$, whereas the dependent variable \widehat{p}_h is the approximation of the trace of p on each face of \mathcal{F}_h . The numerical normal flux (8.4) is characteristic of a particular class of HDG methods, the so-called Local Discontinuous Galerkin Hybridizable (LDG-H) methods proposed and investigated in a series of seminal papers [44, 46, 48, 45].

The quantity $\tau_{\partial K}$ is a nonnegative stabilization parameter and may assume different values on each face $F \in \partial K$ depending on the mesh element which F belongs to. We notice that if $\tau_{\partial K} \simeq \mathcal{O}(h^{-1})$, the second term on the right hand side of (8.4) may be regarded as an artificial diffusion, thus justifying the name *stabilization parameter* attributed to $\tau_{\partial K}$.

Equations (8.5a) and (8.5b) are the HDG discrete counterpart of equations (8.2a) and (8.2b), respectively. These discrete equations hold in the interior of each $K \in \mathcal{T}_h$ and can be solved for each K to eliminate \underline{j}_h^K and p_h^K in favor of the variable $\widehat{p}_h^{\partial K}$. This elimination procedure is called *static condensation* and was introduced for the first time in [59] in the context of dual mixed methods for equilibrium in stress analysis and, subsequently, in [8] in the context of the analysis and implementation of the dual mixed-hybridized finite element method.

Combining this procedure with Eq. (8.4) for each $K \in \mathcal{T}_h$, it is possible to express the normal numerical flux as a function of the sole variable $\widehat{p}_h^{\partial K}$. At this point, we can use the remaining Eqs. (8.5c) and (8.5d). It is convenient to rewrite these equations in a different form by exploiting the property of the test function μ_h of being single-valued on each face of \mathcal{F}_h and discontinuous across face edges. This leads to three sets of distinct equations:

1. one group of equations enforces in a weak sense the interelement continuity of the normal

component of $\widehat{\underline{j}}_h$ across each internal face

$$\langle \llbracket \widehat{\underline{j}}_h \rrbracket, \mu_h \rangle_F = 0, \quad \forall F \in \mathcal{F}_h^0, \quad \mu_h \in M_h; \quad (8.6)$$

2. another group of equations enforces, again in a weak sense, the Dirichlet boundary condition (equation (4.2))

$$\langle \widehat{p}_h, \mu_h \rangle_F = \langle g_D, \mu_h \rangle_F, \quad \forall F \in \mathcal{F}_h^D, \quad \mu_h \in M_h; \quad (8.7)$$

3. another group of equations enforces, again in a weak sense, the Neumann boundary condition (equation (4.3))

$$\langle \widehat{\underline{j}}_h \cdot \underline{n}, \mu_h \rangle_F = \langle g_N, \mu_h \rangle_F, \quad \forall F \in \mathcal{F}_h^N, \quad \mu_h \in M_h; \quad (8.8)$$

4. the last single scalar equation weakly enforcing the integral boundary condition (equation (4.4))

$$\langle \widehat{\underline{j}}_h \cdot \underline{n}, \mu_h \rangle_{\Gamma_I} = I_{target} |\Gamma_I|^{-1} \langle \mu_h, 1 \rangle_{\Gamma_I}, \quad \mu_h \in M_h. \quad (8.9)$$

In [8], it was proven that $\widehat{p}_h|_{\mathcal{F}_h^0}$ can also be interpreted as a *Lagrange multiplier* associated with the continuity condition (8.6). Similarly, we could prove that $\widehat{p}_h|_{\Gamma_I}$ is the multiplier associated with the integral boundary condition (8.9), thereby implying the condition on the left hand side of (4.4). The Dirichlet boundary condition is enforced on the dependent variable \widehat{p}_h via a L^2 projection of g_D as described by Eq. (8.7).

More details about are provided in [13], whereas the implementation is presented in Ch. 9.

2 Error analysis

Let (\underline{j}, p) be the solution of problem defined by Eqs. (4.1a) and (4.1b) with boundary condition described by Eqs. (4.2), (4.3)) and 4.4. We observe that $p \in H^1(\Omega)$, which allows us to define \widehat{p} as the trace of p on the skeleton. In order to prove an error estimate we start with the following consideration: let $P_{M_h} : L^2(\mathcal{F}_h) \rightarrow M_h$ denote the L^2 projection onto M_h . Thanks to the definition of M_h , it is not difficult to realize that the value of $P_{M_h}(\widehat{p})|_{\Gamma_I}$ is the average of \widehat{p} on Γ_I . Since such function is a constant, we have

$$P_{M_h}(\widehat{p}) = \widehat{p} = p \quad \text{on } \Gamma_I. \quad (8.10)$$

In view of this observation the result of Theorem 3.4 in [43] holds with proof unchanged. For the sake of completeness let us verify this by retracing the steps of such a proof.

We start by noticing that the continuous solution $(\underline{j}, p, \widehat{p})$ is sufficiently regular that it verifies, $\forall v_h \in V_h, \forall w_h \in W_h$ and $\forall \mu_h \in M_h$,

$$\sum_{K \in \mathcal{T}_h} [(\mathcal{K}^{-1} \underline{j}, \underline{v}_h^K)_K - (p, \nabla \cdot \underline{v}_h^K)_K + \langle \widehat{p}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K}] = 0, \quad (8.11a)$$

$$\sum_{K \in \mathcal{T}_h} [-(\underline{j}, \nabla w_h^K)_K + \langle \underline{j} \cdot \underline{n}_{\partial K}, w_h^K \rangle_{\partial K}] = \sum_{K \in \mathcal{T}_h} (f, w_h^K)_K, \quad (8.11b)$$

$$\sum_{K \in \mathcal{T}_h} \langle \underline{j} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K} = \langle g_N, \mu_h \rangle_{\Gamma_N} + I_{target} |\Gamma_I|^{-1} \langle \mu_h, 1 \rangle_{\Gamma_I}. \quad (8.11c)$$

Let us introduce an auxiliary space

$$\widehat{M}_h = \{q \in L^2(\mathcal{F}_h) : q|_F \in \mathcal{P}_k(F) \forall F \in \mathcal{F}_h\} \supseteq M_h$$

and let $P_{\widehat{M}_h} : L^2(\mathcal{F}_h) \rightarrow \widehat{M}_h$ denote the L^2 orthogonal projection. We remark that, since p is a constant on Γ_I and identically vanishes on Γ_D , we have $P_{\widehat{M}_h} p = P_{M_h} p$.

Let now define $\mathbf{\Pi} \underline{j} = (\mathbf{\Pi}^K \underline{j})_{K \in \mathcal{T}_h}$, $\mathbf{\Pi} p = (\mathbf{\Pi}^K p)_{K \in \mathcal{T}_h}$ with $\mathbf{\Pi}^K \underline{j} \in \mathbf{P}_k(K)$ and $\mathbf{\Pi}^K p \in \mathcal{P}_k(K)$, as in [43], by

$$(\mathbf{\Pi}^K \underline{j}, \underline{v})_K = (\underline{j}, \underline{v})_K \quad \forall \underline{v} \in \mathbf{P}_{k-1}(K) \quad (8.12a)$$

$$(\mathbf{\Pi}^K p, w)_K = (p, w)_K \quad \forall w \in \mathcal{P}_{k-1}(K) \quad (8.12b)$$

and, $\forall F \in \mathcal{F}_h$ with $F \subset \partial K$,

$$\langle \mathbf{\Pi}^K \underline{j} \cdot \underline{n}_{\partial K} + \tau_{\partial K} \mathbf{\Pi}^K p, \mu \rangle_F = \langle \underline{j} \cdot \underline{n}_{\partial K} + \tau_{\partial K} p, \mu \rangle_F \quad \forall \mu \in \mathcal{P}_k(F). \quad (8.13)$$

We have (Theorem 3.3 of [43])

$$\|\mathbf{\Pi}^K \underline{j} - \underline{j}\|_K \lesssim h_K^{k+1} |\underline{j}|_{k+1, K} + h_K^{k+1} \tau_{\partial K} |p|_{k+1, K}, \quad (8.14a)$$

$$\|\mathbf{\Pi}^K p - p\|_K \lesssim h_K^{k+1} \tau_{\partial K} |p|_{k+1, K} + h_K^{k+1} \tau_{\partial K}^{-1} |\nabla \cdot \underline{j}|_{k+1, K} \quad (8.14b)$$

We let the errors be defined as

$$\begin{aligned} e_{\underline{j}} &= \mathbf{\Pi} \underline{j} - \underline{j}_h & e_p &= \mathbf{\Pi} p - p_h \\ e_{\underline{j}} \cdot \underline{n} &= e_{\underline{j}} \cdot \underline{n} + \tau(e_p - e_{\widehat{p}}) & e_{\widehat{p}} &= P_{\widehat{M}_h} p - \widehat{p}_h \end{aligned}$$

These quantities satisfy the following equations: for $\underline{v}_h \in V_h$, $w_h \in W_h$, and $\mu_h \in M_h \subset \widehat{M}_h$, it holds

$$\sum_{K \in \mathcal{T}_h} \left[(\mathcal{K}^{-1} e_{\underline{j}}^K, \underline{v}_h^K)_K - (e_p^K, \nabla \cdot \underline{v}_h^K)_K + \langle e_{\widehat{p}}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K} \right] = \sum_{K \in \mathcal{T}_h} (\mathcal{K}^{-1} (\mathbf{\Pi}^K \underline{j} - \underline{j}), \underline{v}_h^K)_K, \quad (8.15a)$$

$$\sum_{K \in \mathcal{T}_h} \left[-(e_{\underline{j}}^K, \nabla w_h^K)_K + \langle e_{\underline{j}}^{\partial K} \cdot \underline{n}_{\partial K}, w_h^K \rangle_{\partial K} \right] = 0, \quad (8.15b)$$

$$\sum_{K \in \mathcal{T}_h} \langle e_{\underline{j}}^{\partial K} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K} = 0. \quad (8.15c)$$

Let us prove these equations. By the definition of $\mathbf{\Pi}$, $\mathbf{\Pi}$, and $P_{\widehat{M}_h}$, (8.11a) and (8.11b) imply, on each $K \in \mathcal{T}_h$,

$$\begin{aligned} (\mathcal{K}^{-1} \mathbf{\Pi}^K \underline{j}, \underline{v}_h^K)_K - (\mathbf{\Pi}^K p, \nabla \cdot \underline{v}_h^K)_K + \langle P_{\widehat{M}_h} p, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K} &= (\mathcal{K}^{-1} (\mathbf{\Pi}^K \underline{j} - \underline{j}), \underline{v}_h^K)_K, \\ -(\mathbf{\Pi}^K \underline{j}, \nabla w_h^K)_K + \langle \mathbf{\Pi}^K \underline{j} \cdot \underline{n}_{\partial K} + \tau_{\partial K} (\mathbf{\Pi}^K p - p), w_h^K \rangle_{\partial K} &= (f, w_h^K)_K. \end{aligned}$$

Subtracting the discrete equations (8.5a) and (8.5b) from the above two equations, respectively, and adding over $K \in \mathcal{T}_h$, we obtain (8.15a) and (8.15b). To prove (8.15c), observe that, by definition of $e_{\underline{j}}$, as well as $\mathbf{\Pi}$, $\mathbf{\Pi}$, and $P_{\widehat{M}_h}$, it follows

$$\begin{aligned} \langle e_{\underline{j}}^{\partial K} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K} &= \langle e_{\underline{j}}^K \cdot \underline{n}_{\partial K} + \tau_{\partial K} (e_p^K - e_{\widehat{p}}), \mu_h \rangle_{\partial K} \\ &= \langle (\mathbf{\Pi}^K \underline{j} - \underline{j}_h^K) \cdot \underline{n}_{\partial K} + \tau_{\partial K} (\mathbf{\Pi}^K p - p_h^K - P_{\widehat{M}_h} p + \widehat{p}_h), \mu_h \rangle_{\partial K} \\ &= \langle (\underline{j} - \underline{j}_h^K) \cdot \underline{n}_{\partial K} + \tau_{\partial K} (p - p_h^K - p + \widehat{p}_h), \mu_h \rangle_{\partial K} \\ &= \langle \underline{j} \cdot \underline{n}_{\partial K} - \underline{j}_h^K \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K}. \end{aligned}$$

Adding over all $K \in \mathcal{T}_h$, we get

$$\sum_{K \in \mathcal{T}_h} \langle e_{\underline{j}}^{\partial K} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K} = \sum_{K \in \mathcal{T}_h} \langle \underline{j} \cdot \underline{n}_{\partial K} - \widehat{\underline{j}}_h^{\partial K} \cdot \underline{n}_{\partial K}, \mu_h \rangle_{\partial K}.$$

Since $\underline{j} \in H(\text{div}; \Omega)$ and since $\widehat{\underline{j}}_h$ and \underline{j} satisfy (8.5c) and (8.11c), respectively, the above right hand side is zero.

Finally, taking $v_h = e_{\underline{j}}$ in (8.15a), $w_h = e_p$ in (8.15b) and $\mu_h = -e_{\widehat{p}}$ in (8.15c), and adding the resulting three equations, we obtain

$$\sum_{K \in \mathcal{T}_h} \left[(\mathcal{K}^{-1} e_{\underline{j}}^K, e_{\underline{j}}^K)_K + \langle \tau_{\partial K} (e_p^K - e_{\widehat{p}}), e_p^K - e_{\widehat{p}} \rangle_{\partial K} \right] = \sum_{K \in \mathcal{T}_h} (\mathcal{K}^{-1} (\mathbf{\Pi}^K \underline{j} - \underline{j}), e_{\underline{j}}^K)_K,$$

which is identical to the analogous result in [43], and which, by a standard argument, yields

$$|e_{\underline{j}}|_{0,\Omega}^2 + \sum_{K \in \mathcal{T}_h} \tau_{\partial K} |e_p^K - e_{\widehat{p}}|_{0,\partial K}^2 \lesssim |\mathbf{\Pi} \underline{j} - \underline{j}|_{0,\Omega}^2.$$

The term on the right hand side is bound thanks to (8.14a).

It is not difficult to check that the duality argument used in [47] to bound $\|e_p\|_{0,\Omega}$ holds unchanged also in the present framework. In fact, for w^ϕ solution to $-\Delta w^\phi = \phi$ in Ω , $\phi = 0$ on $\partial\Omega$, the function \widehat{w}_h^ϕ defined on each face F as the $L^2(F)$ projection of $w^\phi|_F$ onto $\mathbb{P}_k(F)$ verifies $\widehat{w}_h^\phi \in M_h$. We then obtain that, if Ω is such that $\phi \in L^2(\Omega)$ implies $w^\phi \in H^2(\Omega)$, we can bound

$$\|e_p\|_{0,\Omega} \lesssim h^{\min\{k,1\}} \|\mathbf{\Pi} \underline{j} - \underline{j}\|_{0,\Omega}.$$

3 Static condensation

In this section, we will be describing how the system (8.4)-(8.5c) can be statically condensed onto a global linear system where only the trace of the solution on the boundaries of the mesh elements shows up. The resulting global system is smaller than that of other DG methods of comparable accuracy [46], and this has resulted in renewed interest in HDG. After solving the global system, the unknowns can be recovered locally, in parallel.

Let us write the space for the numerical trace \widehat{p}_h as

$$M_h = \widetilde{M}_h \oplus M_h^*,$$

with

$$\begin{aligned} \widetilde{M}_h &= \{ \mu \in L^2(\mathcal{F}_h) \mid \mu|_F \in \mathcal{P}_k(F) \quad \forall F \in \mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma N} \cup \mathcal{F}_h^{\Gamma D}, \\ &\quad \mu|_F = 0 \quad \forall F \in \mathcal{F}_h \setminus (\mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma N} \cup \mathcal{F}_h^{\Gamma D}) \} \\ M_h^* &= \{ \mu \in L^2(\mathcal{F}_h) \mid \mu|_F \in \mathbb{R} \quad \forall F \in \mathcal{F}_h^{\Gamma I}, \quad \mu|_F = 0 \quad \forall F \in \mathcal{F}_h \setminus \mathcal{F}_h^{\Gamma I} \}. \end{aligned}$$

Let $\lambda_{1,h} = \widehat{p}_h|_{\widetilde{M}_h}$ and $\lambda_{2,h} = \widehat{p}_h|_{M_h^*}$.

Using integration by parts, the discrete formulation (8.4)-(8.5c) can then be rewritten as follows

$$\sum_{K \in \mathcal{T}_h} \left[(\mathcal{K}^{-1} \underline{j}_h^K, \underline{v}_h^K)_K - (p_h^K, \nabla \cdot \underline{v}_h^K)_K + \langle \lambda_{1,h}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K} + \langle \lambda_{2,h}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K} \right] = 0, \quad (8.16a)$$

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \left[(\nabla \cdot \underline{j}_h^K, w_h^K)_K + \langle \tau_{\partial K} p_h^K, w_h^K \rangle_{\partial K} - \langle \tau_{\partial K} \lambda_{1,h}, w_h^K \rangle_{\partial K} - \langle \tau_{\partial K} \lambda_{2,h}, w_h^K \rangle_{\partial K} \right] \\ = \sum_{K \in \mathcal{T}_h} (f, w_h^K)_K, \end{aligned} \quad (8.16b)$$

$$\sum_{K \in \mathcal{T}_h} [\langle \lambda_{1,h}, \mu_{1,h} \rangle_{\Gamma_D}] = \langle g_D, \mu_{1,h} \rangle_{\Gamma_D}, \quad (8.16c)$$

$$\sum_{K \in \mathcal{T}_h} \left[\langle \underline{j}_h^K \cdot \underline{n}_{\partial K}, \mu_{1,h} \rangle_{\partial K} + \langle \tau_{\partial K} p_h, \mu_{1,h} \rangle_{\partial K} - \langle \tau_{\partial K} \lambda_{1,h}, \mu_{1,h} \rangle_{\partial K} \right] = \langle g_N, \mu_{1,h} \rangle_{\Gamma_N}, \quad (8.16d)$$

$$\sum_{K \in \mathcal{T}_h} [\langle \mathbf{j}_h \cdot \mathbf{n}, \mu_{2,h} \rangle_{\partial K} + \langle \tau_{\partial K} p_h, \mu_{2,h} \rangle_{\partial K} - \langle \tau_{\partial K} \lambda_{2,h}, \mu_{2,h} \rangle_{\partial K}] = I_{target} |\Gamma_I|^{-1} \langle \mu_{2,h}, 1 \rangle_{\Gamma_I}, \quad (8.16e)$$

for all $(\underline{v}_h, w_h, \mu_{1,h}, \mu_{2,h}) \in V_h \times W_h \times \widetilde{M}_h \times M_h^*$.

Observe that the dimension of the space M_h^* is 1, i.e. $M_h^* \cong \mathbb{R}$. As a consequence, (8.16e) is a single scalar equation enforcing the integral boundary condition in a weak sense.

We consider the matrices related to the local bilinear forms

$$A_{11}^K \leftrightarrow (\mathcal{K}^{-1} \underline{j}_h^K, \underline{v}_h^K)_K, \quad A_{12}^K \leftrightarrow (p_h^K, \nabla \cdot \underline{v}_h^K)_K, \quad (8.17a)$$

$$A_{13}^K \leftrightarrow \langle \lambda_{1,h}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K}, \quad A_{14}^K \leftrightarrow \langle \lambda_{2,h}, \underline{v}_h^K \cdot \underline{n}_{\partial K} \rangle_{\partial K}, \quad (8.17b)$$

$$A_{22}^K \leftrightarrow \langle \tau p_h^K, w_h^K \rangle_{\partial K}, \quad A_{23}^K \leftrightarrow \langle \tau \lambda_{1,h}, w_h^K \rangle_{\partial K}, \quad (8.17c)$$

$$A_{24}^K \leftrightarrow \langle \tau \lambda_{2,h}, w_h^K \rangle_{\partial K}, \quad A_f^K \leftrightarrow (f, w_h^K)_K, \quad (8.17d)$$

$$A_{33}^K \leftrightarrow \langle \tau \lambda_{1,h}, \mu_{1,h} \rangle_{\partial K}, \quad A_{44}^K \leftrightarrow \langle \tau \lambda_{2,h}, \mu_{2,h} \rangle_{\partial K}. \quad (8.17e)$$

Let n_V, n_W , and n_M denote the dimensions of $\mathbf{P}_k(K), \mathcal{P}_k(K)$, and $\mathcal{P}_k(F)$, respectively, with $K \in \mathcal{T}_h, F \in \partial K$. Moreover, let NF be the number of faces in $\partial K \cap (\mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N} \cup \mathcal{F}_h^{\Gamma_D})$. Then, it follows

$$\begin{aligned} A_{11}^K &\in \mathbb{R}^{n_V \times n_V}, & A_{12}^K &\in \mathbb{R}^{n_V \times n_W}, & A_{13}^K &\in \mathbb{R}^{n_V \times (\text{NF})n_M}, \\ A_{22}^K &\in \mathbb{R}^{n_W \times n_W}, & A_{23}^K &\in \mathbb{R}^{n_W \times (\text{NF})n_M}, & A_f^K &\in \mathbb{R}^{n_W \times 1}, \\ & & A_{33}^K &\in \mathbb{R}^{(\text{NF})n_M \times (\text{NF})n_M}. \end{aligned}$$

If $\partial K \cap \mathcal{F}_h^{\Gamma_I} \neq \emptyset$, we have

$$A_{14}^K \in \mathbb{R}^{n_V \times 1}, \quad A_{24}^K \in \mathbb{R}^{n_W \times 1}, \quad A_{44}^K \in \mathbb{R}^{1 \times 1} \equiv \mathbb{R},$$

which means that NF is lower than the case without IBC, thus A_{13}^K, A_{23}^K and A_{33}^K are smaller. On the other hand, if $\partial K \cap \mathcal{F}_h^{\Gamma_I} = \emptyset$, the three previous matrices have the expected dimension, whereas are A_{14}^K, A_{24}^K and A_{44}^K are empty.

3.1 Monolithic solver

We now briefly mention the monolithic system to be solved from an algebraic point of view. This global system is impractical an inefficient to solve as-is, the size of the system is indeed much larger than the standard Continuous Galerkin and Discontinuous Galerkin counterparts.

The global system reads

$$\underbrace{\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & 0 \\ A_{41} & A_{42} & 0 & A_{44} \end{pmatrix}}_A \underbrace{\begin{pmatrix} \mathbf{J} \\ P \\ \Lambda_1 \\ \Lambda_2 \end{pmatrix}}_U = \underbrace{\begin{pmatrix} \mathbf{0} \\ A_f \\ 0 \\ 0 \end{pmatrix}}_B \quad (8.18)$$

where the A_{ij}, A_f are given by (8.17a)- (8.17e) $\underline{J}, P, \Lambda_{1,2}$ are the algebraic representation of $\underline{j}_h, p_h, \lambda_{1,h}, \lambda_{2,h}$ respectively. The size of the system is

$$\dim V_h + \dim W_h + \dim M_h = \text{card}\{K\} \times (n_V + n_W) + \text{card}\{\mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N}\} \times \dim n_{\widetilde{M}} + \text{card}\{\mathcal{F}_h^{\Gamma_{\text{bc}}}\} \times \underbrace{\dim n_{M_h^*}}_{=1}.$$

3.2 Local solvers

Consider the following arrays

$$A^K = \begin{bmatrix} A_{11}^K & -A_{12}^K \\ (A_{12}^K)^T & A_{22}^K \end{bmatrix}, \quad B^K = \begin{bmatrix} A_{13}^K & A_{14}^K \\ -A_{23}^K & -A_{24}^K \end{bmatrix}, \quad \underline{F}^K = \begin{bmatrix} 0 \\ A_f^K \end{bmatrix},$$

with $A^K \in \mathbb{R}^{(n_V+n_W) \times (n_V+n_W)}$ and $\underline{F}^K \in \mathbb{R}^{(n_V+n_W)}$. If $\partial K \cap \mathcal{F}_h^{\Gamma_I} \neq \emptyset$, we have $B^K \in \mathbb{R}^{(n_V+n_W) \times ((\text{NF})n_M+1)}$, with $\text{NF} < d+1$; otherwise, $\text{NF} = d+1$, and

$$B^K = \begin{bmatrix} A_{13}^K \\ -A_{23}^K \end{bmatrix} \in \mathbb{R}^{(n_V+n_W) \times ((\text{NF})n_M)}.$$

Eqs. (8.16a)–(8.16b) hold in the interior of each $K \in \mathcal{T}_h$ and can be solved for each K to eliminate \underline{j}_h^K and p_h^K in favor of $\lambda_{1,h}|\partial K$ and $\lambda_{2,h}|\partial K$. Representing $\underline{j}_h^K, p_h^K, \lambda_{1,h}|\partial K$, and $\lambda_{2,h}|\partial K$ with the vectors $\underline{j}_h^K \in \mathbb{R}^{n_V}, p_h^K \in \mathbb{R}^{n_W}, \boldsymbol{\lambda}_1^{\partial K} \in \mathbb{R}^{(\text{NF})n_M}$ and - a possibly empty - scalar $\lambda_2^{\partial K}$, the matrix representation of the local solutions is

$$\begin{bmatrix} \underline{j}_h^K \\ p_h^K \end{bmatrix} = -(A^K)^{-1} B^K \begin{bmatrix} \boldsymbol{\lambda}_1^{\partial K} \\ \lambda_2^{\partial K} \end{bmatrix} + (A^K)^{-1} \underline{F}^K. \quad (8.19)$$

3.3 Flux operators

If $\partial K \cap \mathcal{F}_h^{\Gamma_I} \neq \emptyset$, we define the matrices

$$C^K = \begin{bmatrix} (A_{13}^K)^T & (A_{23}^K)^T \\ (A_{14}^K)^T & (A_{24}^K)^T \end{bmatrix}, \quad D^K = \begin{bmatrix} A_{33}^K & 0 \\ 0 & A_{44}^K \end{bmatrix},$$

where $C^K \in \mathbb{R}^{((\text{NF})n_M+1) \times (n_V+n_W)}$ and $D^K \in \mathbb{R}^{((\text{NF})n_M+1) \times ((\text{NF})n_M+1)}$, with $\text{NF} < d+1$. If $\partial K \cap \mathcal{F}_h^{\Gamma_I} = \emptyset$, $\text{NF} = d+1$ and

$$C^K = \begin{bmatrix} (A_{13}^K)^T & (A_{23}^K)^T \end{bmatrix} \in \mathbb{R}^{((\text{NF})n_M) \times (n_V+n_W)}, \quad D^K = A_{33}^K \in \mathbb{R}^{((\text{NF})n_M) \times ((\text{NF})n_M)}.$$

The flux prescribed by the HDG method

$$\widehat{\underline{j}}_h^{\partial K} = \underline{j}_h^K|_{\partial K} + \tau_{\partial K}(p_h^K|_{\partial K} - \lambda_{1,h}|_{\partial K} - \lambda_{2,h}|_{\partial K}),$$

induces a bilinear form

$$\begin{aligned} \mu_{1,h}, \mu_{2,h} &\rightarrow \langle \widehat{\underline{j}}_h^K \cdot \underline{n}_{\partial K} + \tau_{\partial K}(p_h^K|_{\partial K} - \lambda_{1,h}|_{\partial K} - \lambda_{2,h}|_{\partial K}), \mu_{1,h} + \mu_{2,h} \rangle_{\partial K} \\ &= \langle \underline{j}_h^K \cdot \underline{n}_{\partial K} + \tau_{\partial K} p_h^K|_{\partial K}, \mu_{1,h} + \mu_{2,h} \rangle_{\partial K} \\ &\quad - \langle \tau_{\partial K} \lambda_{1,h}|_{\partial K}, \mu_{1,h} \rangle_{\partial K} - \langle \tau_{\partial K} \lambda_{2,h}|_{\partial K}, \mu_{2,h} \rangle_{\partial K}, \end{aligned}$$

whose matrix representation, using Eq. (8.19), is

$$\begin{aligned} C^K \begin{bmatrix} \underline{j}^K \\ \underline{p}^K \end{bmatrix} - D^K \begin{bmatrix} \lambda_1^{\partial K} \\ \lambda_2^{\partial K} \end{bmatrix} &= -C^K (A^K)^{-1} B^K \begin{bmatrix} \lambda_1^{\partial K} \\ \lambda_2^{\partial K} \end{bmatrix} + C^K (A^K)^{-1} \underline{F}^K - D^K \begin{bmatrix} \lambda_1^{\partial K} \\ \lambda_2^{\partial K} \end{bmatrix} \\ &= \underline{E}_f^K - E^K \begin{bmatrix} \lambda_1^{\partial K} \\ \lambda_2^{\partial K} \end{bmatrix}, \end{aligned} \quad (8.20)$$

with

$$\underline{E}_f^K = C^K (A^K)^{-1} \underline{F}^K, \quad E^K = C^K (A^K)^{-1} B^K + D^K.$$

3.4 Global solver and boundary conditions

Having expressed for each $K \in \mathcal{T}_h$ the numerical flux as a function of $\lambda_{1,h}|_{\partial K}$ and $\lambda_{2,K}|_{\partial K}$ (see Eq. (8.20)), we can use the remaining equations, namely (8.16d) and (8.16e), to actually determine $\lambda_{1,h}$ and $\lambda_{2,h}$. Since the test function $\mu_{1,h} \in \widetilde{M}_h$ is single-valued on each face of \mathcal{F}_h , equation (8.16d) enforces in a weak sense the interelement continuity of the normal component of $\widehat{\underline{j}}_h$ across each internal face

$$\langle \llbracket \widehat{\underline{j}}_h \rrbracket_F, \mu_{1,h} \rangle_F = 0 \quad \forall F \in \mathcal{F}_h^0,$$

and the Neumann boundary condition

$$\langle \widehat{\underline{j}}_h \cdot \underline{n}, \mu_{1,h} \rangle_F = \langle g_N, \mu_{1,h} \rangle_F \quad \forall F \in \mathcal{F}_h^{\Gamma_N}.$$

Similarly, Eq. (8.16e) is a single scalar equation weakly enforcing the integral boundary condition. Eqs. (8.16d) and (8.16e) require the matrices E^K produced element by element to be assembled into a global matrix \mathbb{H} . This matrix collects the fluxes from all the elements, with the result that opposing sign fluxes in internal faces are added. The vectors \underline{E}_f^K also have to be assembled into a global vector \underline{F} .

The global system then reads

$$\mathbb{H} \widehat{\underline{p}} = \underline{F} + \underline{G}_N + \underline{G}_{\Gamma_I}, \quad (8.21)$$

where \underline{G}_N is the vector containing the elements of $\langle g_N, \mu_{1,h} \rangle_{\Gamma_N}$ in the degrees of freedom corresponding to Neumann faces and zeros everywhere else, whereas \underline{G}_{Γ_I} is a vector that has only zeros except for one single entry containing $I_{target} |\Gamma_I|^{-1} \langle \mu_{2,h}, 1 \rangle_{\Gamma_I}$ in the degree of freedom corresponding to faces on $\mathcal{F}_h^{\Gamma_I}$.

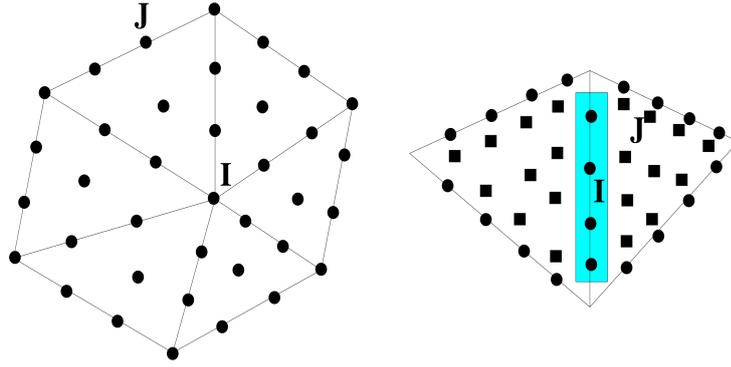


Figure 8.1 – Comparison between the connectivity of the Continuous Galerkin (CG) and HDG methods in the case $d = 2$ without integral boundary conditions. Cubic polynomials are used for both methods. Black circles identify the nodal degrees of freedom of the CG method whereas black circles and black squares identify the degrees of freedom of the HDG method on the boundary and in the interior of each element, respectively. Left panel (CG method): all the elements sharing node I contribute to the connections between node I and the remaining nodes in \mathcal{T}_h . The number of connections (I, J) is equal to 37. Right panel (HDG method): the two neighboring elements are interconnected by the boundary degrees of freedom highlighted in the light blue rectangle. The number of connections (I, J) is equal to 24.

4 Connectivity of the HDG method with IBC

In absence of integral boundary conditions, the HDG method has degrees of freedom inside each element and on each face of \mathcal{F}_h . Therefore, unlike the case of the standard Continuous Galerkin (CG) method, *no* degree of freedom is associated with the mesh vertices. This property makes the HDG method extremely efficient as far as coding is concerned because of its *binary* connectivity structure, i.e., two elements K_1 and K_2 belonging to \mathcal{T}_h share their degrees of freedom - including those in their interiors - *only* if they share a face. This is not the case with the CG method for which nodal continuity at vertices connects the degrees of freedom of a number of elements that is not possible to predict a priori and which depends strongly on the regularity of the mesh. This issue is thoroughly discussed in [128] where a detailed comparison between the computational performance of the CG and HDG methods is carried out in the case $d = 3$. An example of the different connectivity structure associated with the CG and HDG approaches is shown in Figure 8.1 in the case of the space W_h with polynomial degree $k = 3$. Piecewise cubic continuous elements are used also for the CG method. We see that, for the same polynomial degree, the HDG scheme has a far lower connectivity than that of the CG method.

The presence of an integral boundary condition slightly alters the connectivity of the HDG (and CG) method. All the faces $F \in \mathcal{F}_h^{\Gamma_I}$ share a single common degree of freedom $\lambda_{2,h}$. This fact couples all the degrees of freedom on ∂K for all the elements $K \in \mathcal{T}_h$ having, at least, one face $F \in \partial K \cap \mathcal{F}_h^{\Gamma_I}$. Thus, reminding that we denote by n_M the dimension of $P_k(F)$, $F \in \mathcal{F}_h$, the row (column) of global matrix \mathbb{H} corresponding to $\lambda_{2,h}$ will have as many $1 \times n_M$ block rows ($n_M \times 1$ block columns) as the number of faces $F \notin \mathcal{F}_h^{\Gamma_I}$ belonging to the elements coupled by the integral boundary condition, that is, the number of faces in the following set

$$\{F \in \mathcal{F}_h \mid F \in \partial K \setminus \mathcal{F}_h^{\Gamma_I}, \text{ with } \partial K \cap \mathcal{F}_h^{\Gamma_I} \neq \emptyset, \forall K \in \mathcal{T}_h\}.$$

5 Linear elasticity

For simplicity during this chapter we have applied the HDG method to the Darcy equations, however we have applied this discretization method also to the linear elasticity equations (Eq. (5.1)).

Using the same notation introduced in Sec. 1 and following the same steps as done for Darcy equations, we report the final HDG formulation for linear elasticity:

$$\begin{aligned}
(\mathcal{A}\underline{\underline{\sigma}}_h, \underline{v})_{\Omega_h} &+ (\underline{u}_h, \nabla \cdot \underline{v})_{\Omega_h} &- \langle \hat{\underline{u}}_h, \underline{v} \cdot \underline{n} \rangle_{\partial\Omega_h} &= \underline{0} \\
(\nabla \cdot \underline{\underline{\sigma}}_h, \underline{w})_{\Omega_h} &- \left(\rho \frac{\partial^2 \underline{u}_h}{\partial t^2}, \underline{w} \right)_{\Omega_h} - (\tau \underline{\mathbf{P}}_M \underline{u}_h, \underline{w})_{\partial\Omega_h} &+ \langle \tau \hat{\underline{u}}_h, \underline{w} \rangle_{\partial\Omega_h} &= (\underline{F}_{ext}, \underline{w})_{\Omega_h} \\
\langle \underline{\underline{\sigma}}_h \cdot \underline{n}, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &- \langle \tau \mathbf{P}_M \underline{u}_h, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &+ \langle \tau \hat{\underline{u}}_h, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &= 0 \\
\langle \underline{\underline{\sigma}}_h \underline{n}, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &- \langle \tau \mathbf{P}_M \underline{u}_h, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &+ \langle \tau \hat{\underline{u}}_h, \underline{\mu}_1 \rangle_{\partial\Omega_h \setminus \Gamma} &= 0 \\
\langle \underline{\underline{\sigma}}_h \cdot \underline{n}, \underline{\mu}_2 \rangle_{\Gamma_N} &- \langle \tau \mathbf{P}_M \underline{u}_h, \underline{\mu}_2 \rangle_{\Gamma_N} &+ \langle \tau \hat{\underline{u}}_h, \underline{\mu}_2 \rangle_{\Gamma_N} &= \langle \underline{g}_N, \underline{\mu}_2 \rangle_{\Gamma_N} \\
&&&+ \langle \hat{\underline{u}}_h, \underline{\mu}_3 \rangle_{\Gamma_D} &= \langle \underline{g}_D, \underline{\mu}_3 \rangle_{\Gamma_D} \\
\langle \underline{\underline{\sigma}}_h \underline{n}, \underline{\mu}_4 \rangle_{\Gamma_I} &- \langle \tau \mathbf{P}_M \underline{u}_h, \underline{\mu}_4 \rangle_{\Gamma_I} &+ \langle \tau \hat{\underline{u}}_h, \underline{\mu}_4 \rangle_{\Gamma_I} &= \langle \underline{F}_{target}, \underline{\mu}_4 \rangle_{\Gamma_I}
\end{aligned}$$

for all $(\underline{v}, \underline{w}, \underline{\mu}_1, \underline{\mu}_2, \underline{\mu}_3, \underline{\mu}_4) \in \underline{\mathbf{V}}_h \times \underline{\mathbf{W}}_h \times \underline{\mathbf{M}}_h^o \times \underline{\mathbf{M}}_h^N \times \underline{\mathbf{M}}_h^D \times \underline{\mathbf{M}}_h^I$ where

$$\begin{aligned}
\underline{\mathbf{V}}_h &= \left\{ \underline{v} : \Omega \rightarrow \mathbb{R}^{d \times d} : \underline{v}|_K \in [P_k(K)]^{d \times d} \subset \mathbf{S}(K) \quad \forall K \in \Omega_h \right\} \\
\underline{\mathbf{W}}_h &= \left\{ \underline{w} : \Omega \rightarrow \mathbb{R}^d : \underline{w}|_K \in [P_k(K)]^d \quad \forall K \in \Omega_h^d \right\} \\
\underline{\mathbf{M}}_h &= \left\{ \underline{\mu} : \mathcal{F}_h \rightarrow \mathbb{R}^d : \underline{\mu}|_F \in [P_k(F)]^d \quad \forall F \in \mathcal{F}_h \right\} \\
\underline{\mathbf{M}}_h^o &= \left\{ \underline{\mu} \in \underline{\mathbf{M}}_h : \underline{\mu}|_\Gamma = \underline{0} \right\} \\
\underline{\mathbf{M}}_h^D &= \left\{ \underline{\mu} \in \underline{\mathbf{M}}_h : \underline{\mu}|_F = \underline{0} \quad \forall F \in \mathcal{F}_h \setminus \Gamma_D \right\} \\
\underline{\mathbf{M}}_h^N &= \left\{ \underline{\mu} \in \underline{\mathbf{M}}_h : \underline{\mu}|_F = \underline{0} \quad \forall F \in \mathcal{F}_h \setminus \Gamma_N \right\} \\
\underline{\mathbf{M}}_h^I &= \left\{ \underline{\mu} \in \underline{\mathbf{M}}_h : \underline{\mu}|_F = \underline{0} \quad \forall F \in \mathcal{F}_h \setminus \Gamma_I \right\}
\end{aligned}$$

with $\mathbf{S} = \{\text{set of symmetric tensors in } \mathbb{R}^{d \times d}\}$.

Remark 9. Observe that the symmetry for elements $\underline{v} \in \underline{\mathbf{V}}_h$ is enforced in an essential manner in their definition[188].

Here the parameter τ is the stabilization parameter and \mathbf{P}_M denotes the L^2 -orthogonal projection from $\mathbf{L}^2(\mathcal{E}_h)$ onto $\underline{\mathbf{M}}_h$:

$$\hat{\underline{\sigma}}_h \cdot \underline{n} = \underline{\underline{\sigma}}_h \cdot \underline{n} - \tau(\mathbf{P}_M \underline{u}_h - \hat{\underline{u}}_h) \quad \text{on } \partial\Omega_h.$$

An example of application of the HDG formulation with IBC condition for linear elasticity is reported in Ch. 24.

Remark 10. The use of \mathbf{P}_M is needed in order to obtain the optimal accuracy; without it, the formulation might result slightly less precise [188].

Part IV

Contribution to Feel++ Library

In this part we illustrate the implementation of the numerical methods introduced in Part III and the models described in Part II using the Finite Element Embedded Library in C++ (FEEL++) [185]. The purpose is to show how this FEEL++ implementation mirrors the models described in Part II and the methods explained in Part III with a one-to-one correspondence between the math and the C++ code.

This computational framework allows the use of a very wide range of Galerkin and other advanced numerical techniques such as domain decomposition or certified reduced basis. The ingredients include a very expressive embedded language, seamless interpolation and mesh adaption. Moreover, FEEL++ enables parallel computing and allows to solve large scale systems up to tens of thousands of cores. It has been used in various contexts including the development and numerical verification of innovative mathematical methods or the implementation of large multi-physics applications. The range of users span from mechanical engineers in industry, physicists in complex fluids, computer scientists in biomedical applications to applied mathematicians thanks to the shared common mathematical embedded language hiding linear algebra and computer science complexities. FEEL++ provides a mathematical kernel for solving partial differential equation using arbitrary order Galerkin methods (FEM, SEM, CG, DG, CRB) in 1D, 2D, 3D and manifolds using simplices and hypercubes meshes:

- (i) a polynomial library allowing for a wide range polynomial expansions including H_{div} and H_{curl} elements;
- (ii) a light interface to BOOST.UBLAS, EIGEN3 and PETSC/SLEPC as well as a scalable in-house solution strategy;
- (iii) a language for Galerkin methods starting with fundamental concepts such as function spaces, forms, operators, functionals and integrals;
- (iv) a framework that allows user codes to scale seamlessly from single core computation to thousands of cores and enables hybrid computing.

FEEL++ takes advantage of the newest C++ standard (C++ 17) such as type inference and the Boost C++ Libraries such as the BOOST.PARAMETER, BOOST.FUSION, BOOST.HANA or BOOST.MPL and many more. These language enhancements and libraries allow for very concise, robust and expressive C++ codes.

Summarizing, in this part, we present:

- the C++ classes that allows to solve the model using the HDG formulation proposed in Ch. 8;
- the FEEL++ tools we have implemented in order to use the innovative operator splitting illustrated in Ch. 7;
- the C++ classes characterizing the models of the OMVS (Chs. 3 - 5) that allow the user deal with them.

In particular for this last chapter, besides the codes description available for developers, we also show the user friendly interface adapted for a clinical usage, which has been realized in collaboration with the ophthalmologists at the Eye Glick Institute in Indianapolis.

Contents

1	MixedPoisson	87
2	MixedElasticity	92
3	MixedPoissonElasticity	96
4	Computational performances	98

In the following chapter, we discuss the computational framework implementing the numerical method described in Ch. 8, in particular the three C++ classes `MixedPoisson`, `MixedElasticity` and `MixedPoissonElasticity` within the FEEL++ library. These classes are available to be used as toolboxes of FEEL++; at the end of each section we will give some directions how to use these toolboxes *via* the API (Application Programming Interface) provided by FEEL++.

1 MixedPoisson

FEEL++ provides the mathematical language ingredients to implement HDG methods in C++ thanks to its Galerkin Domain Specific Embedded Language (DSEL) for Darcy equations (Eqs. (4.1a) and (4.1b)).

Within the `FeelModels` namespace, we have implemented a C++ class called `MixedPoisson`, child of the superclass `ModelNumerical`. This new template class allows us to use the HDG discretization on Darcy equations and it has been realized with the collaboration of Romain Hild, a PhD student at the University of Strasbourg.

We start introducing some type definitions (Code 9.1) that are very useful in the following,

in particular:

- `Dim` is the domain dimension;
- `Order` is the polynomial order;
- `G_Order` is the polynomial order of the geometric transformation;
- `Pdhv_ptrtype` represents a pointer to a vectorial discontinuous Galerkin space;
- `Pdh_ptrtype` represents a pointer to a scalar discontinuous Galerkin space;
- `Pch_type` represents a pointer to a scalar continuous Galerkin space.

Code 9.1 – Type definition for HDG Darcy

```

1 using mesh_type = Mesh<Simplex<Dim,G_Order>> ;
2 using face_mesh_type = Mesh<Simplex<Dim-1,G_Order,Dim>> ;
3 using Ch_ptr_t = Pch_ptrtype<face_mesh_type,0>;
4 using Vh_ptr_t = Pdhv_ptrtype<mesh_type,Order>;
5 using Wh_ptr_t = Pdh_ptrtype<mesh_type,Order>;
6 using Mh_ptr_t = Pdh_ptrtype<face_mesh_type,Order>;
7 using product2_space_type = ProductSpaces2<Ch_ptr_t,Vh_ptr_t,Wh_ptr_t,Mh_ptr_t>;

```

In Code 9.2 we define the function spaces and meshes.

Code 9.2 – Mesh definition for HDG Darcy

```

1 // load the mesh
2 M_mesh = loadMesh( new mesh_type);

```

Code 9.3 shows the creation of spaces V_h, W_h, \tilde{M}_h and M_h^* . We also want to be able to have several number of integral boundary conditions, the number of which will only be known at execution.

$$V_h \times W_h \times \tilde{M}_h \times (M_h^*)^n$$

Code 9.3 – Function spaces definition for HDG Darcy

```

1 // define the complement of  $\Gamma_{IBC}$  on  $\Gamma$ 
2 auto complement_integral_bdy = complement( faces(M_mesh), [this]( auto const& ewrap ) {
3     auto const& e = unwrap_ref( ewrap );
4     for( auto exAtMarker : this->M_IBCList)
5     {
6         if(e.hasMarker() && e.marker().value()==this->M_mesh->markerName(exAtMarker.marker()))
7             return true;
8     }
9     return false;
10 });
11
12 auto ibc_mesh = createSubmesh(M_mesh,complement_integral_bdy,EXTRACTION_KEEP_MESH_RELATION,0);
13
14 // define function spaces
15 Vh_ptr_t M_Vh = Pdhv<Order>( _mesh=mesh, _extended_dof_table=true );
16 Wh_ptr_t M_Wh = Pdh<Order>( mesh, true );
17 Mh_ptr_t M_Mh = Pdh<Order>( face_mesh,true );
18 // only one degree of freedom
19 Ch_ptr_t M_Ch = Pch<0>(ibc_mesh, true );
20 // define ibc space
21 auto ibcSpaces = std::make_shared<ProductSpace<Ch_ptr_t,true>>( M_integralCondition, M_Ch);
22 // define the product of all spaces
23 M_ps = std::make_shared<product2_space_type>(product2(ibcSpaces,M_Vh,M_Wh,M_Mh));

```

Next we define the linear and bilinear form on M_{ps} , which have each their algebraic representation (Code 9.4).

Code 9.4 – Variational forms definition for HDG Darcy

```

1 solve::strategy s = M_useSC?solve::strategy::static_condensation:solve::strategy::monolithic;
2 M_A = makeSharedMatrixCondensed<double>(s, csrGraphBlocks(*M_ps), *M_backend );
3 M_F = makeSharedVectorCondensed<double>(s, blockVector(*M_ps), *M_backend, false);
4 auto bbf = blockform2(*M_ps, M_A);
5 auto blf = blockform1(*M_ps, M_F);

```

Thus, we assemble the system, in particular we report the implementation of:

- the core of the HDG matrix in Code 9.5;
- the right hand side in Code 9.6;
- the Dirichlet boundary conditions in Code 9.7;
- the Neumann boundary conditions in Code 9.8;
- the integral boundary conditions in Code 9.9.

Code 9.5 – Assemble core of the matrix for HDG Darcy

```

1 // stabilisation parameter
2 auto tau_constant = cst(M_tauCst);
3
4 // static condensation correction parameter
5 auto sc_param = M_useSC ? 0.5 : 1.0;
6
7 //  $\Gamma \setminus \Gamma_{IBC}$ 
8 auto gammaMinusIntegral = complement(boundaryfaces(M_mesh), [this](auto const& ewrap) {
9     auto const& e = unwrap_ref(ewrap);
10    for( auto exAtMarker : this->M_IBCList)
11    {
12        if ( e.hasMarker() && e.marker().value() == this->M_mesh->markerName(exAtMarker.marker()) )
13            return true;
14    }
15    return false;
16 });
17
18 //  $-(p, \nabla \cdot v)_{\Omega}$ 
19 bbf( 0_c, 1_c ) = integrate(_range=elements(M_mesh), _expr=-(idt(p)*div(v)));
20
21 //  $\langle \hat{p}, v \cdot n \rangle_{\Gamma \setminus \Gamma_{IBC}}$ 
22 bbf( 0_c, 2_c ) += integrate(_range=internalfaces(M_mesh),
23     _expr=( idt(phat)*leftface(trans(id(v))*N()) +
24     + idt(phat)*rightface(trans(id(v))*N()) ) );
25 bbf( 0_c, 2_c ) += integrate(_range=gammaMinusIntegral, _expr=idt(phat)*trans(id(v))*N());
26
27 //  $(\nabla \cdot j, q)_{\Omega}$ 
28 bbf( 1_c, 0_c ) += integrate(_range=elements(M_mesh), _expr=- (id(w)*divt(u)));
29
30
31 //  $\langle \tau p, w \rangle_{\Gamma}$ 
32 bbf( 1_c, 1_c ) += integrate(_range=internalfaces(M_mesh),
33     _expr=-tau_constant * ( leftfacet( idt(p) ) * leftface(id(w)) +
34     + rightfacet( idt(p) ) * rightface(id(w)) ) );
35 bbf( 1_c, 1_c ) += integrate(_range=boundaryfaces(M_mesh),
36     _expr=-(tau_constant * id(w)*idt(p)));
37
38 //  $\langle -\tau \hat{p}, w \rangle_{\Gamma \setminus \Gamma_{IBC}}$ 
39 bbf( 1_c, 2_c ) += integrate(_range=internalfaces(M_mesh),
40     _expr=tau_constant * idt(phat) * ( leftface( id(w) ) + rightface( id(w) ) ) );
41 bbf( 1_c, 2_c ) += integrate(_range=gammaMinusIntegral, _expr=tau_constant*idt(phat) * id(w));
42
43 //  $\langle j \cdot n, \mu \rangle_{\Omega \setminus \Gamma}$ 
44 bbf( 2_c, 0_c ) += integrate(_range=internalfaces(M_mesh),
45     _expr=(id(l)*(leftfacet(trans(idt(u))*N())+rightfacet(trans(idt(u))*N())));
46
47 //  $\langle \tau p, \mu \rangle_{\Omega \setminus \Gamma}$ 
48 bbf( 2_c, 1_c ) += integrate(_range=internalfaces(M_mesh),
49     _expr=tau_constant * id(l) * ( leftfacet( idt(p) ) + rightfacet( idt(p) ) ) );
50
51
52 //  $\langle -\tau \hat{p}, \mu \rangle_{\Omega \setminus \Gamma}$ 
53 bbf( 2_c, 2_c ) += integrate(_range=internalfaces(M_mesh),
54     _expr=-sc_param*tau_constant * idt(phat) * id(l) );
55
56 // Here are called the methods to define the boundary conditions within the bilinear form

```

```

57 this->assembleBoundaryCond();
58
59 // Here we assemble the part with the porosity coefficient  $\mathcal{K}$ 
60 for( auto const& pairMat : modelProperties().materials() )
61 {
62     auto marker = pairMat.first;
63     auto material = pairMat.second;
64     if ( !isNL )
65     {
66         auto cond = material.getScalar(M_conductivityKey);
67         //  $(\mathcal{K}^{-1}j, v)_\Omega$ 
68         bbf(0_c, 0_c) += integrate(_quad=_Q<expr_order>(), _range=markedelements(M_mesh,marker),
69                                 _expr=(trans(idt(u))*id(v))/cond );
70     }
71     else
72     {
73         auto cond = material.getScalar(M_n1ConductivityKey, "p", idv(M_pp));
74         //  $(\mathcal{K}(p)^{-1}j, v)_\Omega$ 
75         bbf(0_c, 0_c) += integrate(_quad=_Q<expr_order>(), _range=markedelements(M_mesh,marker),
76                                 _expr=(trans(idt(u))*id(v))/cond );
77     }
78 }
79
80 //  $(\frac{1}{\Delta t}p, w)_\Omega$ 
81 if ( !this->isStationary() ) {
82     bbf( 1_c, 1_c ) += integrate(_range=elements(M_mesh),
83                                 _expr = -(this->timeStepBDF()->polyDerivCoefficient(0)*idt(p)*id(w) );
84 }

```

Code 9.6 – Assemble right hand side for HDG Darcy

```

1 //  $(p_{old}, w)_\Omega$ 
2 if ( !this->isStationary() )
3 {
4     auto bdf_poly = M_bdf_mixedpoisson->polyDeriv();
5     blf(1_c) += integrate(_range=elements(M_mesh), _expr=-inner( idv(bdf_poly), id(w) ) );
6 }
7
8 auto itField = modelProperties().boundaryConditions().find( "potential" );
9 if ( itField != modelProperties().boundaryConditions().end() )
10 {
11     auto mapField = (*itField).second;
12     auto itType = mapField.find( "SourceTerm" );
13     if ( itType != mapField.end() )
14     {
15         for ( auto const& exAtMarker : (*itType).second )
16         {
17             std::string marker = exAtMarker.marker();
18             auto g = expr<expr_order>(exAtMarker.expression());
19             if ( !this->isStationary() )
20                 g.setParameterValues( { "t", M_bdf_mixedpoisson->time() } );
21             blf(1_c) += integrate(_range=markedelements(M_mesh,marker), _expr=-inner(g, id(w)));
22         }
23     }
24 }
25
26 itField = modelProperties().boundaryConditions().find( "flux" );
27 if ( itField != modelProperties().boundaryConditions().end() )
28 {
29     auto mapField = (*itField).second;
30     auto itType = mapField.find( "SourceTerm" );
31     if ( itType != mapField.end() )
32     {
33         for ( auto const& exAtMarker : (*itType).second )
34         {
35             std::string marker = exAtMarker.marker();
36             auto g = expr<Dim, 1, expr_order>(exAtMarker.expression());
37             if ( !this->isStationary() )
38                 g.setParameterValues( { "t", M_bdf_mixedpoisson->time() } );
39             blf(0_c) += integrate(_range=markedelements(M_mesh,marker), _expr=inner(g, id(v) ) );
40         }
41     }
42 }

```

Code 9.7 – Assemble Dirichlet boundary conditions for HDG Darcy

```

1 //  $(\hat{p}_h, \mu)_{\Gamma_{Dirichlet}}$ 
2 bbf( 2_c, 2_c ) += integrate(_range=markedfaces(M_mesh,marker), _expr=idt( phat ) * id(1) );
3
4 //  $(g_D, \mu)_{\Gamma_{Dirichlet}}$ 
5 blf(2_c) += integrate(_range=markedfaces(M_mesh,marker), _expr=id(1) * g);

```

Code 9.8 – Assemble Neumann boundary conditions for HDG Darcy

```

1 // stabilisation parameter
2 auto tau_constant = cst(M_tauCst);
3
4 //  $\langle \mathbf{j} \cdot \mathbf{n}, \mu \rangle_{\Gamma_{Neumann}}$ 
5 bbf( 2_c, 0_c ) += integrate( _range=markedfaces(M_mesh,marker),
6                               _expr=id(1)*(trans(idt(u))*N()) );
7
8 //  $\langle \tau \hat{p}, \mu \rangle_{\Gamma_{Neumann}}$ 
9 bbf( 2_c, 1_c ) += integrate( _range=markedfaces(M_mesh,marker),
10                              _expr=tau_constant*id(1)*idt(p) );
11
12 //  $\langle -\tau \hat{p}, \mu \rangle_{\Gamma_{Neumann}}$ 
13 bbf( 2_c, 2_c ) += integrate( _range=markedfaces(M_mesh,marker),
14                              _expr=-tau_constant*idt(phant)*id(1) );
15
16 //  $\langle g_N, \mu \rangle_{\Gamma_{Neumann}}$ 
17 blf(2_c) += integrate( _range=markedfaces(M_mesh, marker), _expr=id(1) * g);

```

Code 9.9 – Assemble integral boundary conditions for HDG Darcy

```

1 //  $\langle \lambda, \mathbf{v} \cdot \mathbf{n} \rangle_{\Gamma_{IBC}}$ 
2 bbf( 0_c, 3_c, 0, i ) += integrate( _range=markedfaces(M_mesh,marker),
3                                     _expr= idt(uI)*(trans(id(u))*N()) );
4
5 //  $\langle \lambda, \tau w \rangle_{\Gamma_{IBC}}$ 
6 bbf( 1_c, 3_c, 1, i ) += integrate( _range=markedfaces(M_mesh,marker),
7                                     _expr=tau_constant*idt(uI)*id(w) );
8
9 //  $\langle \mathbf{j} \cdot \mathbf{n}, \mu \rangle_{\Gamma_{IBC}}$ 
10 bbf( 3_c, 0_c, i, 0 ) += integrate( _range=markedfaces(M_mesh,marker),
11                                     _expr=(trans(idt(u))*N()) * id(nu) );
12
13 //  $\langle \tau \hat{p}, \mu \rangle_{\Gamma_{IBC}}$ 
14 bbf( 3_c, 1_c, i, 1 ) += integrate( _range=markedfaces(M_mesh,marker),
15                                     _expr=tau_constant *idt(p)*id(nu) );
16
17 //  $\langle -\lambda, \mu \rangle_{\Gamma_{IBC}}$ 
18 bbf( 3_c, 3_c, i, i ) += integrate( _range=markedfaces(M_mesh,marker),
19                                     _expr=-tau_constant*id(nu)*idt(uI) );
20
21 double meas = integrate( _range=markedfaces(M_mesh,marker), _expr=cst(1.0).evaluate()(0,0);
22 //  $\langle I_{target}, \mu \rangle_{\Gamma_{IBC}}$ 
23 blf(3_c,i) += integrate( _range=markedfaces(M_mesh,marker), _expr=g*id(nu)/meas );

```

$0_c, 1_c, 2_c \dots$ refer to the index at compile time of the test or trial function spaces in M_{ps} . Hence $\text{bbf}(0_c, 0_c)$ refers to the terms associated with the flux.

We can choose at runtime if the problem is solved using static condensation or the monolithic way. Although the later becomes rapidly inefficient in 3D with respect to the mesh size, it is useful to provide this construction the debug and verify the static condensation process.

The last step consists in solving the problem using either the monolithic or static condensation strategy and retrieve the components (flux and potential) of the solution field (Code 9.10). The local solves, to reconstruct the flux and potential once the trace is obtained, are taking advantage of the embarrassingly parallel context using multi-threading, including within the MPI process when we run in parallel.

Code 9.10 – Solve the system for HDG Darcy

```

1 auto U = M_ps->element();
2
3 // static condensation is done during the solve
4 bbf.solve(_solution=U, _rhs=blf, _condense=M_useSC, _name=prefix());
5
6 // get views over each component
7 M_up = U(0_c); // element of  $V_h$ 
8 M_pp = U(1_c); // element of  $W_h$ 
9 for( int i = 0; i < M_integralCondition; i++ )
10   M_mup[i] = U(3_c,i); // elements of  $M_h^*$ 

```

We list here the main methods that are available in the API provided by feel in order to use this toolbox:

- `mesh()`, which allows to access to the computational mesh;

- `potentialSpace()`, `fluxSpace()`, `traceSpace()`, which allow to access to the primal, dual and trace spaces, respectively;
- `potentialField()`, `fluxField()`, which allow to access to the primal and dual variables, respectively;
- `init(mesh, meshVisu)`, which initialize the overall framework (mesh, BCs, initial conditions, spaces, exporter). The user can pass directly the computational and the visualization mesh, otherwise it is automatically loaded;
- `assembleAll()`, which assembles both the constant part (`assembleCstPart()`) and non-constant part (`assembleNonCstPart()`) of the Darcy system. In particular, the constant assembler method builds the core of the matrix including the part of BCs that are in the matrix and don't change at each time step. On the other hand, the non-constant assembler method adds to the matrix the time-dependent and nonlinear contributions, the part of BCs that are missing - Robin - and builds the right hand side of the Darcy system;
- `solve()`, which solves the system matrix-right hand side built using `assembleAll()`;
- `exportResults(time, mesh, Idh, Idhv)`, which exports the results at the time-step `time` on the visualization mesh `mesh` using the interpolation operators for the primal and dual variables provided by `Idh` and `Idhv`, respectively.

2 MixedElasticity

In this section we present the `MixedElasticity` class. Similarly to the previous one, thank to the powerful tools provided by FEEL++, we have implemented the HDG method for linear elasticity equations (Sec. 5, Ch. 8) in C++. Also in this case we have implemented the class `MixedElasticity` within the `FeelModels` namespace as child of the superclass `ModelNumerical`.

We start introducing some type definitions (Code 9.11) that are very useful in the following, in particular:

- `Dim` is the domain dimension;
- `Order` is the polynomial order;
- `G_Order` is the geometrical order;
- `Pdhms_ptrtype` represents a pointer to a symmetric matrix discontinuous Galerkin space;
- `Pdhv_ptrtype` represents a pointer to a vectorial discontinuous Galerkin space;
- `Pdh_ptrtype` represents a pointer to a scalar discontinuous Galerkin space;
- `Pchv_type` represents a pointer to a vectorial continuous Galerkin space.

Code 9.11 – Type definition for HDG elasticity

```

1 using mesh_type = Mesh<Simplex<Dim,G_Order>> ;
2 using face_mesh_type = Mesh<Simplex<Dim-1,G_Order,Dim>> ;
3 using Ch_ptr_t = Pchv_ptrtype<face_mesh_type,0>;
4 using Vh_ptr_t = Pdhms_ptrtype<mesh_type,Order>;
5 using Wh_ptr_t = Pdhv_ptrtype<mesh_type,Order>;
6 using Mh_ptr_t = Pdhv_ptrtype<face_mesh_type,Order>;
7 using product2_space_type = ProductSpaces2<Ch_ptr_t,Vh_ptr_t,Wh_ptr_t,Mh_ptr_t>;

```

Next, Codes 9.12 and 9.13 illustrates the process of loading a new mesh and creating new function spaces, respectively.

Code 9.12 – Mesh definition for HDG elasticity

```

3 // load the mesh
4 M_mesh = loadMesh( new mesh_type);

```

Code 9.13 – Function spaces definition for HDG elasticity

```

1 // Mh only on the faces whitout integral condition
2 auto complement_integral_bdy = complement( faces(M_mesh), [this]( auto const& e ) {
3     for( auto exAtMarker : this->M_IBCList)
4     {
5         if ( e.marker().value() == this->M_mesh->markerName( exAtMarker.marker() ) )
6             return true;
7     }
8     return false;
9 });
10
11 auto face_mesh=createSubmesh(M_mesh,complement_integral_bdy,EXTRACTION_KEEP_MESH_RELATION,0);
12 std::vector<std::string> ibc_markers(M_integralCondition);
13 for( int i = 0; i < M_integralCondition; i++)
14     ibc_markers.push_back(M_IBCList[i].marker());
15 auto ibc_mesh = createSubmesh( M_mesh, markedfaces(M_mesh, ibc_markers),
16     EXTRACTION_KEEP_MESH_RELATION, 0 );
17
18 // define function spaces
19 M_Vh = Pdhms<Order>( M_mesh, true );
20 M_Wh = Pdhv<Order>( M_mesh, true );
21 M_Mh = Pdhv<Order>( face_mesh, true );
22 M_M0h = Pdh<0>( face_mesh );
23 M_Ch = Pchv<0>( ibc_mesh, true );
24
25 // define ibc space
26 auto ibcSpaces = std::make_shared<ProductSpace<Ch_ptr_t,true>>( M_integralCondition, M_Ch);
27
28 // define the product of all spaces
29 M_ps = std::make_shared<product2_space_type>(product2(ibcSpaces,M_Vh,M_Wh,M_Mh));

```

Then, in Code 9.14 we define the algebraic representations of the linear and bilinear form based on M_{ps} .

Code 9.14 – Variational forms definition for HDG elasticity

```

1 solve::strategy s = M_useSC?solve::strategy::static_condensation:solve::strategy::monolithic;
2 M_A = makeSharedMatrixCondensed<double>(s, csrGraphBlocks(*M_ps), *M_backend );
3 M_F = makeSharedVectorCondensed<double>(s, blockVector(*M_ps), *M_backend, false);
4 auto bbf = blockform2( *M_ps, M_A);
5 auto blf = blockform1( *M_ps, M_F );

```

Finally, as before, we assemble the system:

- in Code 9.15 the core of the HDG elasticity matrix;
- in Code 9.16 the right hand side;
- in Code 9.17 the Dirichlet boundary conditions ;
- in Code 9.18 the Neumann boundary conditions;
- in Code 9.19 the integral boundary conditions.

Code 9.15 – Assemble core of the matrix for HDG elasticity

```

1 auto gammaMinusIntegral = complement(boundaryfaces(M_mesh), [this]( auto const& e ) {
2   for( auto exAtMarker : this->M_IBCList )
3     {
4       if ( e.marker().value() == this->M_mesh->markerName( exAtMarker.marker() ) )
5         return true;
6     }
7   return false;
8 });
9
10 auto sc_param = M_useSC ? 0.5 : 1.0;
11 for( auto const& pairMat : modelProperties().materials() )
12 {
13   auto material = pairMat.second;
14   auto lambda = material.getScalar("lambda");
15   auto mu = material.getScalar("mu");
16   auto c1 = cst(0.5)/mu;
17   auto c2 = -lambda/(cst(2.) * mu * (cst(Dim)*lambda + cst(2.)*mu));
18
19   //  $(A\sigma, v)_\Omega$ 
20   bbf(0_c, 0_c) += integrate(_range=elements(M_mesh), _expr=(c1*inner(idt(sigma), id(v))));
21   bbf(0_c, 0_c) += integrate(_range=elements(M_mesh), _expr=(c2*trace(idt(sigma))*trace(id(v))));
22 }
23
24 //  $(u, \nabla \cdot v)_\Omega$ 
25 bbf( 0_c, 1_c ) += integrate(_range=elements(M_mesh), _expr=(trans(idt(u))*div(v)));
26
27 //  $(-\hat{u}, v \cdot n)_\Omega \setminus \Gamma$ 
28 bbf( 0_c, 2_c ) += integrate(_range=internalfaces(M_mesh), _expr=-(trans(idt(uhat))*
29   leftface(id(v)*N()) + trans(idt(uhat))*rightface(id(v)*N())));
30
31 //  $(-\hat{u}, v \cdot n)_\Gamma \setminus \Gamma_{IBC}$ 
32 bbf( 0_c, 2_c ) += integrate(_range=gammaMinusIntegral, _expr=-trans(idt(uhat))*(id(v)*N()) );
33
34 //  $(\nabla \cdot \underline{\sigma}, w)_\Omega$ 
35 bbf( 1_c, 0_c ) += integrate(_range=elements(M_mesh), _expr=(trans(id(w))*divt(sigma)));
36
37 if ( !this->isStationary() )
38 {
39   auto dt = this->timeStep();
40   for( auto const& pairMat : modelProperties().materials() )
41     {
42       auto material = pairMat.second;
43       auto rho = material.getScalar("rho");
44       //  $(-\rho \frac{d^2 u}{dt^2}, w)_\Omega$ 
45       bbf( 1_c, 1_c ) += integrate(_range=elements(M_mesh), _expr=-rho*inner(idt(u), id(w))/(dt*dt));
46     }
47 }
48
49 //  $(-\tau u, w)_\Omega \setminus \Gamma$ 
50 bbf( 1_c, 1_c ) += integrate(_range=internalfaces(M_mesh),
51   _expr=-tau_constant * ( leftfacet( trans(idt(u)) ) * leftface(id(w)) +
52   rightfacet( trans(idt(u)) ) * rightface(id(w)) ) );
53
54 //  $(-\tau u, w)_\Gamma$ 
55 bbf( 1_c, 1_c ) += integrate(_range=boundaryfaces(M_mesh), _expr=-tau_constant * trans(idt(u)) * id(w));
56
57 //  $(\tau \hat{u}, w)_\Omega \setminus \Gamma$ 
58 bbf( 1_c, 2_c ) += integrate(_range=internalfaces(M_mesh),
59   _expr=tau_constant * ( leftfacet( trans(idt(uhat)) ) * leftface(id(w)) +
60   rightfacet( trans(idt(uhat)) ) * rightface(id(w)) ) );
61
62 //  $(\tau \hat{u}, w)_\Gamma \setminus \Gamma_{IBC}$ 
63 bbf( 1_c, 2_c ) += integrate(_range=gammaMinusIntegral, _expr=tau_constant * trans(idt(uhat)) * id(w));
64
65 //  $(\underline{\sigma} \cdot n, \mu_1)_\Omega \setminus \Gamma$ 
66 bbf( 2_c, 0_c ) += integrate(_range=internalfaces(M_mesh),
67   _expr=(trans(id(m)) * (leftfacet(idt(sigma)*N()) + rightfacet(idt(sigma)*N())) ) );
68
69 //  $(-\tau u, \mu_1)_\Omega \setminus \Gamma$ 
70 bbf( 2_c, 1_c ) += integrate(_range=internalfaces(M_mesh),
71   _expr=-tau_constant * trans(id(m)) * (leftfacet(idt(u)) + rightfacet(idt(u))));
72
73 //  $(\tau \hat{u}, \mu_1)_\Omega \setminus \Gamma$ 
74 bbf( 2_c, 2_c ) += integrate(_range=internalfaces(M_mesh),
75   _expr=sc_param * tau_constant * trans(idt(uhat)) * id(m));

```

Code 9.16 – Assemble right hand side for HDG elasticity

```

1 //  $(u_{old}, w)_\Omega$ 
2 if ( !this->isStationary() )
3 {
4   for( auto const& pairMat : modelProperties().materials() )

```

```

5     {
6       auto material = pairMat.second;
7       auto rho = material.getScalar("rho");
8       auto u = this->timeStepNM()->previousUnknown(0);
9       auto u1 = this->timeStepNM()->previousUnknown(1);
10      auto dt = this->timeStep();
11
12      blf(1_c) += integrate(_range=elements(M_mesh),
13                          _expr=-rho*inner(2*idv(u)-idv(u1), id(w))/(dt*dt));
14    }
15  }
16
17  auto itField = modelProperties().boundaryConditions().find("stress");
18  if (itField != modelProperties().boundaryConditions().end() )
19  {
20    auto mapField = (*itField).second;
21    auto itType = mapField.find("SourceTerm");
22    if ( itType != mapField.end() )
23    {
24      for ( auto const& exAtMarker : (*itType).second )
25      {
26        auto g = expr<Dim,1,expr_order> (exAtMarker.expression());
27        if ( !this->isStationary() )
28          g.setParameterValues( { {"t", M_nm_mixedelasticity->time()} } );
29        blf( 1_c ) += integrate(_range=elements(M_mesh), _expr=trans(g)*id(w));
30      }
31    }
32  }

```

Code 9.17 – Assemble Dirichlet boundary conditions for HDG elasticity

```

1 //  $\langle \hat{u}, \mu_1 \rangle_{\Gamma_{Dirichlet}}$ 
2 bbf( 2_c, 2_c ) += integrate(_range=markedfaces(M_mesh,marker), _expr=trans(idt(uhat)) * id(m) );
3
4 //  $\langle \hat{g}_D, \mu_1 \rangle_{\Gamma_{Dirichlet}}$ 
5 bbf( 2_c ) += integrate(_range=markedfaces(M_mesh,marker), _expr=trans(id(m))*g);

```

Code 9.18 – Assemble Neumann boundary conditions for HDG elasticity

```

1 //  $\langle \hat{\sigma} \cdot \hat{n}, \mu_1 \rangle_{\Gamma_{Neumann}}$ 
2 bbf(2_c, 0_c) += integrate(_range=markedfaces(M_mesh,marker),
3                          _expr=(trans(id(m)) * (idt(sigma)*N())));
4
5 //  $\langle -\tau \hat{u}, \mu_1 \rangle_{\Gamma_{Neumann}}$ 
6 bbf(2_c, 1_c) += integrate(_range=markedfaces(M_mesh,marker),
7                          _expr=-tau_constant*trans(id(m))*idt(u));
8
9 //  $\langle \tau \hat{u}, \mu_1 \rangle_{\Gamma_{Neumann}}$ 
10 bbf(2_c, 2_c) += integrate(_range=markedfaces(M_mesh,marker),
11                          _expr=tau_constant*trans(idt(uhat))*id(m));
12
13 //  $\langle \hat{g}_N, \mu_1 \rangle_{\Gamma_{Neumann}}$ 
14 bbf(2_c) += integrate(_range=markedfaces(M_mesh,marker), _expr=trans(id(m))*g);

```

Code 9.19 – Assemble integral boundary conditions for HDG elasticity

```

1 //  $\langle -\hat{u}_I, \hat{v} \cdot \hat{n} \rangle_{\Gamma_{IBC}}$ 
2 bbf(0_c, 3_c, 0, 1) += integrate(_range=markedfaces(M_mesh,marker),
3                                _expr=-trans(idt(uI)) * (id(v)*N()));
4
5 //  $\langle \tau \hat{u}_I, \hat{w} \rangle_{\Gamma_{IBC}}$ 
6 bbf(1_c, 3_c, 1, i) += integrate(_range=markedfaces(M_mesh,marker),
7                                _expr=tau_constant*trans(idt(uI)));
8
9 //  $\langle \hat{\sigma} \cdot \hat{n}, \mu_2 \rangle_{\Gamma_{IBC}}$ 
10 bbf(3_c, 0_c, i, 0) += integrate(_range=markedfaces(M_mesh,marker),
11                                _expr=inner(idt(v)*N(), id(nu)));
12
13 //  $\langle \tau \hat{u}_I, \mu_2 \rangle_{\Gamma_{IBC}}$ 
14 bbf(3_c, 1_c, i, 1) += integrate(_range=markedfaces(M_mesh,marker),
15                                _expr=tau_constant*inner(idt(u), id(nu)));
16
17 //  $\langle -\tau \hat{u}_I, \mu_2 \rangle_{\Gamma_{IBC}}$ 
18 bbf(3_c, 3_c, i, i) += integrate(_range=markedfaces(M_mesh,marker),
19                                _expr=-tau_constant*inner(idt(uI), id(nu)));
20
21 double meas = integrate(_range=markedfaces(M_mesh,marker), _expr=cst(1.0)).evaluate()(0,0);
22 //  $\langle F_{target}, \mu_2 \rangle_{\Gamma_{IBC}}$ 
23 bbf(3_c, i) += integrate(_range=markedfaces(M_mesh,marker), _expr=inner(g, id(nu))/meas);

```

As for the `MixedPoisson` class, the user can always choose between the monolithic solver

or exploit the computational power of the static condensation. The solve method is shown in Code 9.20.

Code 9.20 – Solve the system for HDG elasticity

```

1 auto U = M_ps->element();
2
3 // static condensation is done during the solve
4 bbf.solve(_solution=U, _rhs=blf, _rebuild=false, _condense=M_useSC, _name= this->prefix());
5
6 // get views over each component
7 M_up = U(0_c); // element of  $V_h$ 
8 M_pp = U(1_c); // element of  $W_h$ 
9 for( int i = 0; i < M_integralCondition; i++ )
10 M_mup[i] = U(3_c,i); // elements of  $M_h^*$ 

```

We list here the main methods that are available in the API provided by feel in order to use this toolbox:

- `mesh()` allows to access to the computational mesh;
- `displacementSpace()`, `stressSpace()`, `traceSpace()` allow to access to the primal, dual and trace spaces, respectively;
- `fieldDisplacement()`, `fieldStress()` allow to access to the primal and dual variables, respectively;
- `init(mesh, meshVisu)` initializes the overall framework (mesh, BCs, initial conditions, spaces, exporter). The user can pass directly the computational and the visualization mesh, otherwise it is automatically loaded;
- `assembleCstPart()` assembles the matrix of the linear elasticity system, including the matrix contributions due to BCs;
- `assembleNonCstPart()` assembles the right hand side of the linear elasticity system;
- `solve()` solves the system matrix-right hand side built using `assembleCstPart()` and `assembleNonCstPart()`;
- `exportResults(time, mesh, Idh, Idhv)` exports the results at the time-step time on the visualization mesh `mesh` using the interpolation operators for the primal and dual variables provided by `Idh` and `Idhv`, respectively.
- `exportTimers()` exports in a text file the computational times needed by the program to complete each step in assembling and solving the linear elasticity system.

3 MixedPoissonElasticity

In this section we present the `MixedPoissonElasticity` class. This class is an interface between the `MixedPoisson` and the `MixedElasticity` classes and it allows to solve the system defined by Eq. (5.13).

This class has mainly two attributes that corresponds to an element of `MixedPoisson` and one of `MixedElasticity`. Thus, we implemented two methods in order to modify the HDG formulation in order to take into account the coupling terms between Darcy and linear elasticity.

Code 9.21 – Extra terms in the Darcy formulation

```

1 //  $\langle -\frac{d}{dt} \nabla \cdot \underline{u}, w \rangle_{\Omega}$ 
2 blf(1_c) += integrate( _range=elements( M_mesh ),
3   _expr= -( div(M_ElasticityModel-> potentialField()) +
4     - div(M_ElasticityModel->timeStepNM()->previousUnknown()) * id(w)/dt );

```

Code 9.21 shows the elastic contribution to the right hand side of Eq. 4.1b, whereas Code 9.22 illustrates the Darcy contribution to the linear elastic system.

Code 9.22 – Extra terms in the linear elasticity formulation

```

1 auto pressure = M_PoissonModel->potentialField();
2
3 //  $\langle p \underline{I}, v \rangle_{\Omega}$ 
4 blf( 0_c ) += integrate( _range=elements( M_mesh ),
5   _expr= - inner( idv(pressure)*eye<Dim,Dim>(), id(v) );
6
7 // Adding extra-term for special Neumann
8 //  $\langle -p \underline{I} \cdot \underline{n}, \underline{\mu} \rangle_{\Gamma_{Neumann}}$ 
9 std::string marker = "";
10 for( auto const& pairMat : M_ElasticityModel->modelProperties().materials() )
11 {
12   auto material = pairMat.second;
13   marker = material.getString("special_neumann");
14 }
15 if (!marker.empty())
16   blf( 2_c ) += integrate( _range=markedfaces(M_mesh,marker),
17     _expr= - inner( idv(pressure)*eye<Dim,Dim>() * N(), id(m) );

```

4 Computational performances

In this section we briefly discuss the computational cost of the different model implemented in FEEL++. The computational domain used in this section is displayed in Fig. 9.1.

These performance studies have been completed on one of the five nodes of ATLAS, the computing resource cluster at Cemosis and IRMA laboratory. The node used has 24 cores on 2 sockets (Intel Xeon E5-2680 v3 2.50GHz) hyperthreaded with 256 GB of RAM . Everything is interconnected with both 10Gb Ethernet cards and 40Gb Infini-band cards. The workload manager is slurm*.

The solvers used are retrieved from PETSc [1], a suite of data structures and routines for scalable and parallel solution of scientific applications modeled by partial differential equations. To partition the domain for parallel computing we exploit METIS [122]. METIS is a set of serial programs for partitioning graphs, partitioning finite element meshes, and producing fill reducing orderings for sparse matrices. The algorithms implemented in METIS are based on the multilevel recursive-bisection, multilevel k-way, and multi-constraint partitioning schemes developed in the Department of Computer Science & Engineering at the University of Minnesota.

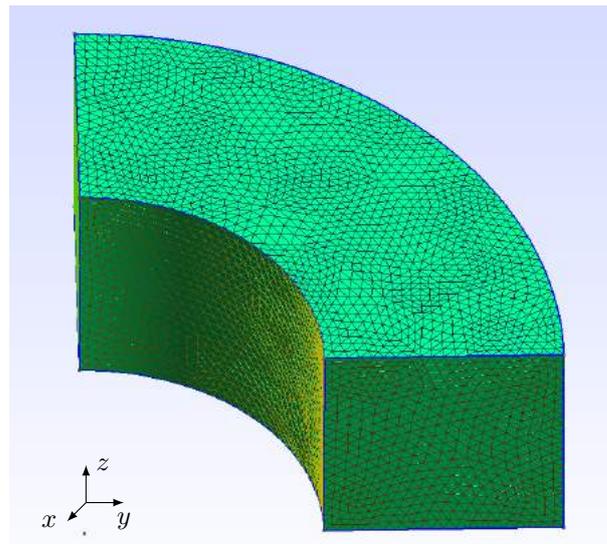


Figure 9.1 – Computational domain used for the performance studies.

Darcy

For the simple Darcy - only Neumann and Dirichlet boundary conditions - we have performed three test cases using the toolbox `MixedPoisson` with increasing number of degrees of freedom (dofs) as described by Tab. 9.1. Insuring the same computing error for the results, we compare the time needed to solve the Darcy system in a monolithic way with two different preconditioners - LU factorization [6] and the Generalized Additive Schwarz Method (GASM) [5] s- and the computational time required by the static condensation using the Geometric Al-

*<https://slurm.schedmd.com/documentation.html>

gebraic MultiGrid (GAMG) preconditioner [5]. The analytical solution used for the pressure is $\frac{10}{\pi} \tan(y, x)^{-1}$ and for the velocity is $\underline{v} = [-\frac{10}{\pi} \frac{y}{x^2 + y^2}, \frac{10}{\pi} \frac{x}{x^2 + y^2}, 0]^T$.

TEST CASE	DOFS PRIMAL VARIABLE	DOFS DUAL VARIABLE	DOFS TRACE VARIABLE
<i>Darcy 1</i>	137436	45812	72981
<i>Darcy 2</i>	1054884	351628	543792
<i>Darcy 3</i>	2047728	682576	1049316

Table 9.1 – Data for the Darcy performance study.

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Darcy 1</i>	Monolithic	LU	16.33841	11.4856	7.622147
<i>Darcy 2</i>	Monolithic	LU	1050.858	681.0828	404.7277
<i>Darcy 3</i>	Monolithic	LU	5469.657	4241.529	2017.625
<i>Darcy 1</i>	Monolithic	GASM	34.26085	21.4217	7.329376
<i>Darcy 2</i>	Monolithic	GASM	2249.435	757.5647	127.7822
<i>Darcy 3</i>	Monolithic	GASM	7670.616	4147.575	444.0716
<i>Darcy 1</i>	Static condensation	GAMG	8.858283	5.636978	4.447856
<i>Darcy 2</i>	Static condensation	GAMG	77.54065	46.61523	26.65524
<i>Darcy 3</i>	Static condensation	GAMG	180.9639	108.0637	56.79263

Table 9.2 – Wall computational time of the Darcy performance study. np : number of processors used to run the test case.

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Darcy 1</i>	Monolithic	LU	7.607082	5.55147	3.372929
<i>Darcy 2</i>	Monolithic	LU	984.8325	637.2967	378.456
<i>Darcy 3</i>	Monolithic	LU	5335.235	4145.453	1961.585
<i>Darcy 1</i>	Monolithic	GASM	19.97991	12.16076	3.21067
<i>Darcy 2</i>	Monolithic	GASM	2149.023	697.6506	99.76536
<i>Darcy 3</i>	Monolithic	GASM	7530.162	4024.747	386.3492
<i>Darcy 1</i>	Static condensation	GAMG	2.926887	1.19151	0.8900033
<i>Darcy 2</i>	Static condensation	GAMG	34.13834	19.44127	9.769887
<i>Darcy 3</i>	Static condensation	GAMG	80.2323	48.85748	20.95092

Table 9.3 – Solver computational time of the Darcy performance study. np : number of processors used to run the test case.

Tab. 9.2 shows the wall time using 1, 2 or 4 processors by the monolithic solver and the static condensation. The results obtained suggest that the static condensation technique increases the speed up proportionally to the dimension of the problem, for instance up to 35 times faster in the case of *Darcy 3*. Looking at the solver time (Tab. 9.3), this boost is even more important (more than 90 times faster in the case of *Darcy 3* with 4 processors). We haven't reported the time for the assembly of the matrix but they are equivalent in all the cases studied.

Darcy with IBC

For the Darcy with IBC, we have performed three test cases using `MixedPoisson` with increasing number of degrees of freedom (dofs) as described by Tab. 9.4. Insuring again that the computing error is the same for all the test cases, we compare the time needed to solve the Darcy system in a monolithic way with LU and GASM as preconditioners, and the computational time required by the static condensation using the Geometric Algebraic MultiGrid (GAMG) preconditioner. The analytical solution used for the pressure is $\frac{10}{\pi} \tan(y, x)^{-1}$ and for the velocity is $\underline{v} = [-\frac{10}{\pi} \frac{y}{x^2 + y^2}, \frac{10}{\pi} \frac{x}{x^2 + y^2}, 0]^T$.

TEST CASE	DOFS PRIMAL VARIABLE	DOFS DUAL VARIABLE	DOFS TRACE VARIABLE
<i>Darcy + IBC 1</i>	137436	45812	72249
<i>Darcy + IBC 2</i>	1054884	351628	540606
<i>Darcy + IBC 3</i>	2048016	682672	1045050

Table 9.4 – Data for the Darcy with IBC performance study.

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Darcy + IBC 1</i>	Monolithic	LU	17.545	11.969	7.70089
<i>Darcy + IBC 2</i>	Monolithic	LU	1143.502	853.1877	379.6678
<i>Darcy + IBC 3</i>	Monolithic	LU	10043.3	5949.22	2002.92
<i>Darcy + IBC 1</i>	Monolithic	GASM	33.666	24.1658	8.11918
<i>Darcy + IBC 2</i>	Monolithic	GASM	2348.37	760.623	121.776
<i>Darcy + IBC 3</i>	Monolithic	GASM	7886.05	3300.11	440.802
<i>Darcy + IBC 1</i>	Static condensation	GAMG	8.898762	6.22261	4.52851
<i>Darcy + IBC 2</i>	Static condensation	GAMG	79.01226	47.30553	29.8194
<i>Darcy + IBC 3</i>	Static condensation	GAMG	166.223	111.875	68.9631

Table 9.5 – Wall computational time of the Darcy with IBC performance study. np : number of processors used to run the test case.

Tab. 9.5 shows the wall time using 1, 2 or 4 processors by the monolithic solver and the static condensation. The results obtained suggest that the static condensation technique increases the speed up proportionally to the dimension of the problem, for instance up to 40 times faster in the case of *Darcy + IBC 3* with respect to monolithic with LU. Looking at the solver time (Tab. 9.6), this boost is even more important (100 times faster in the case of *Darcy + IBC 3* in serial). We haven't reported the time for the assembly of the matrix but they are equivalent in all the cases studied.

In this case we found that if we increase the number of processors the standard partitioning is slowing down any solver, due to the fact that the dofs of the IBC are shared by multiple processors and the number of communications becomes the bottleneck of the computation. We call *special* partitioning the case in which we are imposing to the partitioner that the IBC dofs are all assigned to the same processor, seeking for a more balanced partition of the computational work. We consider that each IBC face is a neighbor of all the other IBC faces, which put a huge weight on the connectivity pattern of the IBC faces and METIS keeps them on the same domain

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Darcy + IBC 1</i>	Monolithic	LU	8.62135	6.06181	3.501
<i>Darcy + IBC 2</i>	Monolithic	LU	1078.169	809.579	352.9749
<i>Darcy + IBC 3</i>	Monolithic	LU	9855.69	5830.8	1945.71
<i>Darcy + IBC 1</i>	Monolithic	GASM	19.4689	14.8323	3.75853
<i>Darcy + IBC 2</i>	Monolithic	GASM	2250.7	702.701	94.3866
<i>Darcy + IBC 3</i>	Monolithic	GASM	7706.46	3176.52	384.504
<i>Darcy + IBC 1</i>	Static condensation	GAMG	3.089028	2.07635	1.21391
<i>Darcy + IBC 2</i>	Static condensation	GAMG	36.02654	23.08251	12.85595
<i>Darcy + IBC 3</i>	Static condensation	GAMG	82.531	54.0843	30.4405

Table 9.6 – Solver computational time of the Darcy with IBC performance study. np : number of processors used to run the test case.

when partitioning. In this way we get back appropriate scalability in Darcy problems with IBC.

We performed 4 test cases using the static condensation with GAMG as preconditioner, which are described by Tab. The analytical solution used for the pressure is $\frac{10}{\pi} \tan(y, x)^{-1}$ and for the velocity is $\underline{v} = [-\frac{10}{\pi} \frac{y}{x^2 + y^2}, \frac{10}{\pi} \frac{x}{x^2 + y^2}, 0]^T$.

TEST CASE	DOFS PRIMAL VARIABLE	DOFS DUAL VARIABLE	DOFS TRACE VARIABLE
<i>IBC partitioning 1</i>	137436	45812	72249
<i>IBC partitioning 2</i>	1054164	351388	540606
<i>IBC partitioning 3</i>	2046192	682064	1044138
<i>IBC partitioning 4</i>	16140492	5380164	8153232

Table 9.7 – Data for the IBC partitioning study.

TEST CASE	PARTITIONING	TIME [s]			
		$np = 2$	$np = 4$	$np = 8$	$np = 16$
<i>IBC partitioning 1</i>	standard	6.29754	5.32076	5.5787	4.93148
<i>IBC partitioning 2</i>	standard	78.8595	31.8898	87.1904	137.479
<i>IBC partitioning 3</i>	standard	167.966	68.7134	327.265	384.741
<i>IBC partitioning 4</i>	standard	2148.14	1007.48	17573.3	22119.8
<i>IBC partitioning 1</i>	special	10.4937	4.89363	4.46361	5.2293
<i>IBC partitioning 2</i>	special	71.9122	34.8672	26.5368	26.167
<i>IBC partitioning 3</i>	special	142.188	73.8138	52.9313	60.1039
<i>IBC partitioning 4</i>	special	2076.27	956.377	785.842	575.329

Table 9.8 – Wall computational time of the IBC partitioning study. np : number of processors used to run the test case.

In Tabs. 9.8 and 9.9 we report the computational time for this study. As theoretically predicted when we increase the number of processors, especially for larger problems, the computational time is even increasing instead of scaling in the case of standard partitioning. With

TEST CASE	PARTITIONING	TIME [s]			
		$np = 2$	$np = 4$	$np = 8$	$np = 16$
<i>IBC partitioning 1</i>	standard	1.97274	1.74082	2.16775	1.94948
<i>IBC partitioning 2</i>	standard	38.9133	14.3937	72.1542	127.225
<i>IBC partitioning 3</i>	standard	97.5774	32.8774	298.298	364.875
<i>IBC partitioning 4</i>	standard	1132.78	412.666	17147.3	21778.5
<i>IBC partitioning 1</i>	special	3.77704	1.30827	0.965129	1.40304
<i>IBC partitioning 2</i>	special	34.5642	15.0314	10.9773	10.7097
<i>IBC partitioning 3</i>	special	73.9961	32.6929	23.2586	26.0804
<i>IBC partitioning 4</i>	special	1145.71	424.482	354.697	231.272

Table 9.9 – Solver computational time of the IBC partitioning study. np : number of processors used to run the test case.

the simple proposed special partitioning we obtain good results, indeed the largest problem with 16 processors and the special partitioning is solved 38 times faster than the standard partitioning problem and the solver itself is 94 times faster. We haven't reported the time for the assembly of the matrix but they are equivalent in all the cases studied.

Elasticity

For the elasticity problem, we have performed three test cases using `MixedElasticity` with increasing number of degrees of freedom (dofs) as described by Tab. 9.10. Insuring again that the computing error is the same for all the test cases, we compare the time needed to solve the elastic system in a monolithic way with LU as preconditioner, and the computational time required by the static condensation using the `fieldsplit` preconditioner. The analytical solution used for the displacement is $\underline{u} = [x^2 + y^2 + z^2, x^2 + y^2 + z^2, x^2 + y^2 + z^2]^T$.

TEST CASE	DOFS PRIMAL VARIABLE	DOFS DUAL VARIABLE	DOFS TRACE VARIABLE
<i>Elasticity 1</i>	374928	187464	299259
<i>Elasticity 2</i>	1216608	608304	952677
<i>Elasticity 3</i>	2327880	1163940	1808298

Table 9.10 – Data for the elasticity performance study.

Tab. 9.11 shows the wall time using 1, 2 or 4 processors by the monolithic solver and the static condensation. The results obtained suggest that the static condensation technique increases the speed up proportionally to the dimension of the problem also for elasticity, for instance up to 25 times faster in the case of *Elasticity 3* with 2 processors with respect to monolithic. Looking at the solver time (Tab. 9.12), this boost is even more important (31 times faster in the case of *Elasticity 3* with 2 processors). We haven't reported the time for the assembly of the matrix but they are equivalent in all the cases studied.

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Elasticity 1</i>	Monolithic	LU	87.43	74.09	36.09
<i>Elasticity 2</i>	Monolithic	LU	793.23	694.30	238.14
<i>Elasticity 3</i>	Monolithic	LU	3707.38	2956.88	663.74
<i>Elasticity 1</i>	Static condensation	fieldsplit	28.46	16.64	10.09
<i>Elasticity 2</i>	Static condensation	fieldsplit	104.81	59.76	30.83
<i>Elasticity 3</i>	Static condensation	fieldsplit	214.47	116.74	61.15

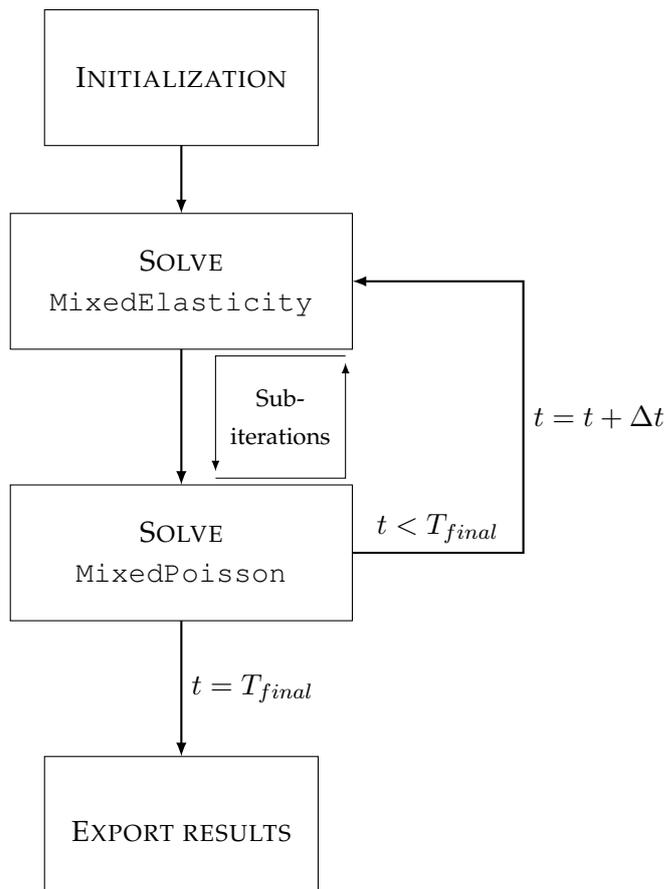
Table 9.11 – Wall computational time of the elasticity performance study. np : number of processors used to run the test case.

TEST CASE	SOLVER	PRECONDITIONER	TIME [s]		
			$np = 1$	$np = 2$	$np = 4$
<i>Elasticity 1</i>	Monolithic	LU	50.02	43.52	20.82
<i>Elasticity 2</i>	Monolithic	LU	665.61	563.73	155.10
<i>Elasticity 3</i>	Monolithic	LU	3458.89	2632.42	522.84
<i>Elasticity 1</i>	Static condensation	fieldsplit	18.71	10.46	5.89
<i>Elasticity 2</i>	Static condensation	fieldsplit	73.68	41.58	19.76
<i>Elasticity 3</i>	Static condensation	fieldsplit	150.32	82.66	41.06

Table 9.12 – Solver computational time of the elasticity performance study. np : number of processors used to run the test case.

Poroelastic system

The solver framework for the poroelastic system is illustrated by Fig 9.2. First, we initialize both the `MixedPoisson` and `MixedElasticity` classes, therefore, we start solving the linear elasticity system passing the information to the Darcy equations and sub-iterating this process till the error between two sub-iterations is below a certain tolerance or we have attained the maximum number of sub-iterations. Then, we move to the next time-step and we repeat the process till we reach the final time, when we export the results and exit the program. For the solver and the exporter, we exploit the methods available for each subclass (`MixedPoisson` and `MixedElasticity`).

Figure 9.2 – Solver framework for `MixedPoissonElasticity` class.

3D-0D operator splitting method

Contents

1	Functional Mock-up Interface(FMI)	105
2	FEEL++ and Functional Mock-up Unit (FMU)	106
3	The operator splitting in FEEL++	107

In this chapter we present the implementation of the coupling algorithm explained in Ch. 7 in FEEL++.

First we give a brief introduction of the Functional Mock-up Interface (FMI) and its employment in FEEL++. Next we describe the C++ implemented version of the operator splitting algorithm. Finally, we illustrate the results of a convergence study performed using the afore-said framework developed in FEEL++.

1 Functional Mock-up Interface(FMI)

The FMI is a modeling environment that describes a product sub-system by differential, algebraic and discrete equations with time, state and step-events. The power of these models that are widely used in industry, is the easiness in the interconnection with other libraries. This interface generates and export the modeling components as Functional Mock-up Units (FMUs). These units can be imported and used in embedded control systems and, moreover, can cooperate at runtime through a co-simulation environment. For further details on FMI and FMU, we refer to [18].

For our purpose we have implemented the 0D model (Ch. 3) in OpenModelica [76], an open-source Modelica-based modeling and simulation environment. We have adopted this

open-source software due to its user-friendly interface and its simplicity to develop complex electric circuits, in order to focus on the modeling framework and not on the numerical methods to solve lumped-parameter models that have been largely studied in literature.

From OpenModelica it is possible to export a FMU that fully describes the circuit model. In particular, such export is organized in one zip-file containing several files:

- an XML-file that contains the definition of all variables in the model and other information, such as solver parameters;
- a small set of easy to use C-functions, which define all the needed model equations;
- other optional data such as documentation, maps and tables.

2 FEEL++ and Functional Mock-up Unit (FMU)

In order to use the FMU in FEEL++, we implement an interface class called `FMU`. This work has been done in collaboration with J.-B. Wahl, a research engineer at the Université de Strasbourg.

We have imported in FEEL++ the following methods *via* a wrapper that exploit the C functions already defined within the FMILIB library:

- `int load(std::string _path)` loads the model in the FMU format from the input directory path;
- `void initialize(double t_init, double t_final, double tolerance)` initializes the model with the input initial and final time, the input solver tolerance, and the default parameter values specified in the XML-file within the FMU;
- `template <typename VariableType> void setValue(std::string var_name, VariableType value)` allows the user to set the value of a variable of the FMU model at runtime;
- `template <typename VariableType> VariableType getValue(std::string var_name)` allows the user to retrieve the value of a variable of the FMU model at runtime;
- `double currentTime()` returns the value of the time of the current simulation;
- `void reset()` resets the simulation time to the initial value and reset all the settings to their default value;
- `void simulate(double t_init, double t_final, double tolerance)` runs the model from the initial time to final time;
- `void doSteps(double t_stop)` runs the model from the current simulation time to a specified simulation time.

We report just the main functions that are useful for the simulation, however there are other methods implemented needed by the interface FMILIB-FEEL++ for a more advanced usage. All the source codes are available and open to be accessed on the GitHub repository at <https://github.com/feelpp>.

3 The operator splitting in FEEL++

In order to use in FEEL++ the operator splitting method presented in Ch. 7, we implemented the class `CoupledMixedPoisson` child of `MixedPoisson`. In particular this new class has the following attributes:

- `Ch_element_vector_type M_Y` is a vector that contains the buffer variables;
- `Bdf_element_vector_t M_bdf_buffer` is a the corresponding vector of the buffer variables that handles the time dependency and BDF algorithm;
- `int M_0dCondition` is an integer that indicates the number of 0D circuit for the coupling
- `integral_boundary_list_type M_0dList` collects the interface markers between the 3D and the 0D circuits;
- `fmu_ptr_vector_t M_circuit` is a pointer vector that gathers the 0D circuits.

The initialization of the variables and the spaces exploits the methods already defined for `MixedPoisson`, the only complication is the allocation of the new spaces for the buffer variable that has to be take into account (Code 10.1)

Code 10.1 – Allocation of new spaces for the operator splitting class

```

1  if(M_0dCondition)
2  {
3      auto zeroSpaces = std::make_shared<ProductSpace<Ch_ptr_t,true>>
4          (this->integralCondition()+M_0dCondition, this->M_Ch);
5      this->M_ps = std::make_shared<product2_space_type>
6          (product2(zeroSpaces,this->M_Vh,this->M_Wh,this->M_Mh));
7      solve::strategy s = boption(prefixvm(this->prefix(), "use-sc"))?
8          solve::strategy::static_condensation:solve::strategy::monolithic;
9
10     this->M_A_cst = makeSharedMatrixCondensed<value_type>
11         (s, csrGraphBlocks(*(this->getPS())), *(this->get_backend()) );
12     this->M_A = makeSharedMatrixCondensed<value_type>
13         (s, csrGraphBlocks(*(this->getPS())), *(this->get_backend()) );
14     this->M_F = makeSharedVectorCondensed<value_type>
15         (s, blockVector(*(this->getPS())), *(this->get_backend()), false);
16
17     M_bdf_buffer.resize( M_0dCondition );
18 }
19
20 // Initialization of the buffer variable
21 for (int i = 0; i < M_0dCondition; i++)
22     M_Y.push_back(this->constantSpace()->element( "yy" ));
23
24 // Initialization for second step: the 0d equation
25 if (M_0dCondition)
26 {
27     M_circuit = std::make_shared<FMU>();
28     M_circuit->load();
29     M_circuit->initialize( this->timeInitial(), this->timeFinal() );
30 }

```

Code 10.2 shows the assembly of the new contribution in the HDG matrix due to the coupling with the 0D circuit. In detail, this piece of C++ code extracted from the FEEL++ library,

shows that for matrix line $3_c, i$ we have imposed a similar homogeneous IBC condition on Γ_I as described in Sec. 1, Ch. 9. Tab. 10.1 depicts a scheme of the 3D HDG matrix with 0D coupling condition, in particular the *Step 1* of the operator splitting. In this figure we have boxed the novel contributions with respect to the HDG with IBC formulation.

$(\underline{\mathcal{K}}^{-1} \underline{j}, \underline{v})_\Omega$	$(-p, \nabla \cdot \underline{v})_\Omega$	$\langle \hat{p}, \underline{v} \cdot \underline{n} \rangle_{\Gamma_{int} \cup \Gamma \setminus \Gamma_{IBC}}$	$\langle p_I, \underline{v} \cdot \underline{n} \rangle_{\Gamma_I}$
$(\nabla \cdot \underline{j}, q)_\Omega$	$\left(\frac{\partial p}{\partial t}, w \right)_\Omega + (\tau p, w)_{\Gamma_{int} \cup \Gamma}$	$-\langle \tau \hat{p}, w \rangle_{\Gamma_{int} \cup \Gamma \setminus \Gamma_I}$	$\langle \tau p_I, w \rangle_{\Gamma_I}$
$\langle \underline{j} \cdot \underline{n}, \underline{\mu}_1 \rangle_{\Gamma_{int} \cup \Gamma_N}$	$\langle \tau p, \underline{\mu}_1 \rangle_{\Gamma_{int} \cup \Gamma_N}$	$-\langle \tau \hat{p}, \underline{\mu}_1 \rangle_{\Gamma_{int} \cup \Gamma_N} + \langle \tau \hat{p}, \underline{\mu}_1 \rangle_{\Gamma_D}$	
$\langle \underline{j} \cdot \underline{n}, \underline{\mu}_2 \rangle_{\Gamma_I}$	$\langle \tau p, \underline{\mu}_2 \rangle_{\Gamma_I}$		$-\tau p_I \boxed{-\frac{1}{R} \frac{1}{ \Gamma_I } p_I, \underline{\mu}_2}_{\Gamma_I}$
			$\boxed{\frac{1}{R} \frac{1}{ \Gamma_I } \langle \Pi, \underline{\mu}_2 \rangle_{\Gamma_I}}$
			$\boxed{-\frac{1}{R} \frac{1}{ \Gamma_I } \langle \underline{u}_I, \underline{\mu}_3 \rangle_{\Gamma_I}}$
			$\boxed{\langle \frac{1}{R} \frac{1}{ \Gamma_I } Y + \frac{C}{ \Gamma_I } \frac{Y}{\Delta t}, \underline{\mu}_3 \rangle_{\Gamma_I}}$

Table 10.1 – Scheme of *Step 1* of the 3D HDG formulation with 0D circuit coupling. In particular the boundary $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_I$ where Γ_D , Γ_N , and Γ_I are the Dirichlet, Neumann boundary conditions, and integral interface condition, respectively. Γ_{int} corresponds to the internal faces and $(\underline{v}, w, \underline{\mu}_1, \underline{\mu}_2, \underline{\mu}_3)$ are the test functions for the flux, potential, trace, IBC and 0D space, respectively. Finally, Π is the unknown variable on the resistor-capacitor buffer (resistance R and capacitance C). We have boxed the contributions due to the coupling with a 0D circuit.

Code 10.2 – Matrix contribution due to 0D coupling

```

1 // stabilisation parameter
2 auto tau_constant = cst(doption(prefixvm(this->prefix(), "tau_constant")));
3
4 auto marker = M_0dList[i].marker();
5 int j = i+1;
6
7 // 0D EQUATION
8 double meas=integrate(_range=markedfaces(this->mesh(),marker),_expr=cst(1.0)).evaluate()(0,0);
9 double Cbuffer = 1; double Rbuffer = 1;
10 std::string Cbuffer_str; std::string Rbuffer_str;
11 for( auto const& pairMat : this->modelProperties().materials() )
12 {
13     auto material = pairMat.second;
14     Cbuffer_str = material.getString( "Cbuffer_name" );
15     Cbuffer_str += ".C";
16     Cbuffer = M_circuit->getValue<double>( Cbuffer_str ); // Capacitance of the buffer
17     Rbuffer_str = material.getString( "Rbuffer_name" );
18     Rbuffer_str += ".R";
19     Rbuffer = M_circuit->getValue<double>( Rbuffer_str ); // Resistance of the buffer
20 }
21
22 // Part on the IBC line
23 //  $-\frac{1}{R} \frac{1}{|\Gamma_I|} \langle p_I, \underline{\mu}_2 \rangle_{\Gamma_I}$ 
24 bbf ( 3_c, 3_c, i, i ) += integrate( _range=markedfaces(this->mesh(),marker),
25     _expr = -idt(uI)*id(nu)/Rbuffer/meas );
26
27 //  $\frac{1}{R} \frac{1}{|\Gamma_I|} \langle Y, \underline{\mu}_2 \rangle_{\Gamma_I}$ 
28 bbf ( 3_c, 3_c, i, j ) += integrate( _range=markedfaces(this->mesh(),marker),
29     _expr = idt(yy)*id(nu)/Rbuffer/meas );
30
31 // Part on the 0d line
32 //  $-\frac{1}{R} \frac{1}{|\Gamma_I|} \langle p_I, \underline{\mu}_3 \rangle_{\Gamma_I}$ 
33 bbf ( 3_c, 3_c, j, i ) += integrate( _range=markedfaces(this->mesh(),marker),
34     _expr = -idt(uI)*id(nu)/Rbuffer/meas );
35
36 //  $\frac{1}{R} \frac{1}{|\Gamma_I|} \langle Y, \underline{\mu}_3 \rangle_{\Gamma_I}$ 
37 bbf ( 3_c, 3_c, j, j ) += integrate( _range=markedfaces(this->mesh(),marker),
38     _expr = idt(yy)*id(nu)/Rbuffer/meas );
39
40 //  $\langle \frac{C}{|\Gamma_I|} \frac{Y}{\Delta t}, \underline{\mu}_3 \rangle_{\Gamma_I}$ 
41 bbf ( 3_c, 3_c, j, j ) += integrate( _range=markedfaces(this->mesh(),marker),
42     _expr= Cbuffer*M_bdf_buffer[i]->polyDerivCoefficient(0)*idt(yy)*id(nu)/meas );

```

Remark 11. Unfortunately the static condensation has been only very recently available in FEEL++ in order to solve the first step of the proposed 3D-0D coupling algorithm. The difference with respect to the standard or IBC case is the presence of extra entries in the matrix that derive from the coupling with the ODEs system. Nevertheless we have estimated the computational time needed to solve an unsteady Darcy+ODEs problem using the monolithic algorithm. For a test case using monolithic solver and LU

preconditioner with 10 time steps, 223512 degrees of freedom (dofs) for the primal variable, 112092 dofs for the dual variable, and 176139 dofs for the trace, the FEEL++ code takes 759.3847 s and 68.27 s in serial and in parallel with 10 processors, respectively.

OMVS implementation

Contents

1	FEEL++ interface	111
1.1	PatientAssessment	112
1.2	Configuration file	114
1.3	Darcy json file	115
1.4	Elasticity json file	117
1.5	Computational performances	118
2	OMVS user friendly interface	118
2.1	OMVS simulation interface	119
2.2	OMVS View App	125
2.3	OMVS Compare App	126

This chapter is divided into two sections: first we present the C++ codes that are implemented in FEEL++ in order to deal with the OMVS, in particular the organization and how to use the model we have developed; second, we show the user friendly interface we have implemented such that also researchers that are not mathematicians, computer scientist, engineers or not familiar with C++, can use the OMVS for scientific research and as a clinical tool in ophthalmologic centers.

1 FEEL++ interface

We discuss now the interface between the models described in Chs. 3, 4 and 5 and the user *via* FEEL++. In this case we assume that the user can be also a developer and has some expertise in coding or to deal with C++ software or run program on Linux based shells.

This interface is based on a C++ class called `PatientAssessment` where the data from the input files are collected and passed to the models implemented in FEEL++.

1.1 PatientAssessment

First, we present this C++ class that allows the user handle the data to feed the model. In particular `PatientAssessment` has the attributes resumed in Tab. 11.1, whereas Tab. 11.2 shows the methods of this class.

NAME	DESCRIPTION
<code>std::string M_date</code>	date of the assessment
<code>int M_level</code>	version of the OMVS
<code>pt::ptree M_pt</code>	pointer to the properties of the model
<code>ptr_model M_model</code>	pointer to the FMU file that contains the 0D model
<code>double M_SBP</code>	assessment systolic blood pressure
<code>double M_DBP</code>	assessment diastolic blood pressure
<code>double M_HR</code>	assessment heart rate
<code>double M_IOP</code>	assessment intraocular pressure
<code>double M_RLTp</code>	assessment translaminar tissue pressure
<code>double M_CRVp</code>	assessment central retinal vein pressure
<code>double M_CRAp</code>	assessment central artery vein pressure
<code>std::string M_note</code>	assessment note

Table 11.1 – Attributes of the class `PatientAssessment`.

The `PatientAssessment` class, however, is not sufficient to handle all the different type of entries that an end-user would like to have for this kind of application, thus, we implemented the class `Patient` that manages the data directly from the input files to the models combined with the class presented above.

The C++ class `Patient` is able to collect multiple assessments of the same subject and handle therefore the patient clinical history.

The different assessments are stored in `map`, which is an associative container that stores elements formed by a combination of a key value and a mapped value, following a specific order. In our case the key is created from the date of the assessment and the value is the collection of the patient clinical data and the output computed by the OMVS. The attributes of this class, accompanied by the corresponding set and get methods, are:

- `std::string M_name` stores the name of the patient;
- `int M_age` stores the patient age;
- `Gender M_gender` stores the patient gender (undefined, male, female);
- `bool M_upload` tells if the patient has already been uploaded on the online private repository;
- `M_override` specifies if the new assessments have to override the results already uploaded on the online private repository;

NAME	DESCRIPTION
<code>std::string now() const</code>	returns a string containing the current date and time
<code>void setDate(std::string const& d)</code>	sets the date <code>d</code> or, if <code>d</code> is not specified, embed the current date within the model for the analyzed patient
<code>std::string const& date() const</code>	returns the date of the analyzed assessment
<code>void setLevel(int l)</code>	sets the level of assessment with $l = -1, 0, 1, 2$, $l = -1$ means we run all possible models
<code>int level() const;</code>	returns the assessment level
<code>bool hasLevel0() const</code>	returns <code>True</code> if <i>Level 0</i> is available, <code>False</code> otherwise
<code>bool hasLevel1() const</code>	returns <code>True</code> if <i>Level 1</i> is available, <code>False</code> otherwise
<code>bool hasLevel2() const</code>	returns <code>True</code> if <i>Level 2</i> is available, <code>False</code> otherwise
<code>void setNote(std::string const& n)</code>	allows the user set a note for the assessment
<code>std::string const& note() const</code>	returns the assessment note
<code>pt::ptree const& pTree() const</code>	returns the property of the model
<code>void fromJson(std::string const& s)</code>	reads the input data file string and load the information into patient
<code>void setSystolicBloodPressure(double SBP)</code>	sets the assessment systolic blood pressure
<code>double systolicBloodPressure() const</code>	returns the assessment systolic blood pressure
<code>void setDiastolicBloodPressure(double DBP)</code>	sets the assessment diastolic blood pressure
<code>double diastolicBloodPressure() const</code>	returns the assessment diastolic blood pressure
<code>void setMeanHeartRate(double HR)</code>	sets the assessment average heart rate
<code>double meanHeartRate() const</code>	returns the assessment average heart rate
<code>void setIntraOcularPressure(double IOP)</code>	sets the assessment intraocular pressure
<code>double intraOcularPressure() const</code>	returns the assessment intraocular pressure
<code>void setRetroLaminarTissuePressure(double RLTP)</code>	sets the assessment retrolaminar tissue pressure
<code>double retroLaminarTissuePressure() const</code>	returns the assessment retrolaminar tissue pressure
<code>void setCentralRetinaVeinPressure(double CRVp)</code>	sets the assessment central retinal vein pressure
<code>double centralRetinaVeinPressure() const</code>	returns the assessment central retinal vein pressure
<code>void setCentralRetinaArteryPressure(double CRAp)</code>	sets the assessment central artery vein pressure
<code>double centralRetinaArteryPressure() const</code>	returns the assessment central artery vein pressure
<code>void run()</code>	run the assessment

Table 11.2 – Methods of the class `PatientAssessment`.

- `std::string M_note` stores notes of the patient;
- `std::shared_ptr<PatientAssessment> M_assessment` is the map that stores all the different assessments of the same patient.

Fig. 11.1 illustrates the data processing from input to output. At the beginning the data are specified by the user *via* text files, then are passed through classes `Patient` and `PatientAssessment` that handle the interface with the FEEL++ Eye2Brain library. This library exploits the implementations of HDG (Ch. 9) and the operator splitting (Ch. 10) and manages the different levels of the OMVS in order to finally provide the simulation results to the user.

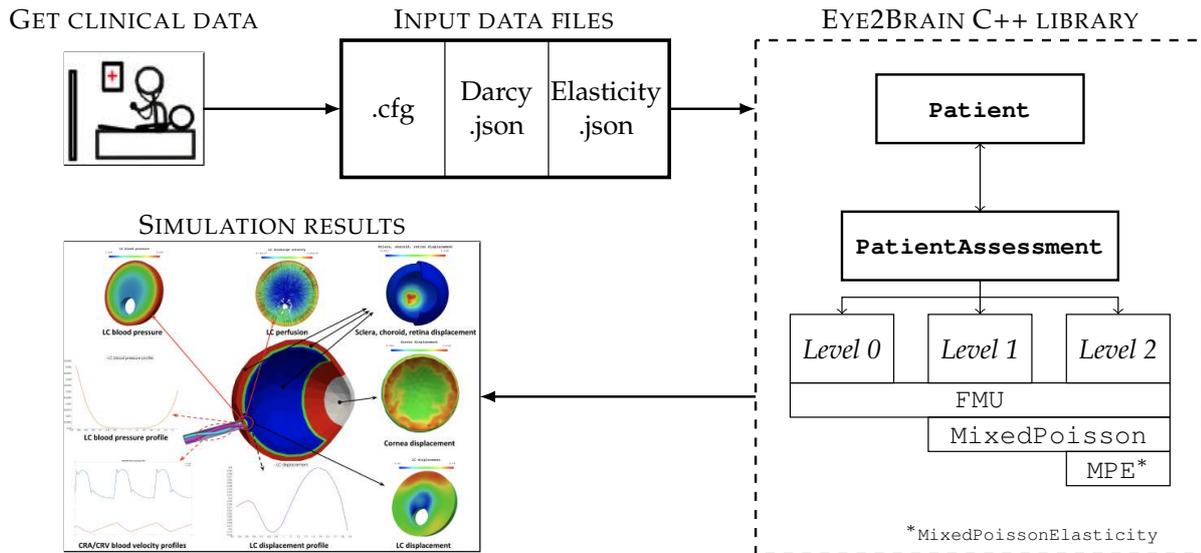


Figure 11.1 – FEEL++ data processing from input to output.

1.2 Configuration file

The configuration file is the input text file that is needed by any version of the OMVS in order to interact directly with the Eye2Brain library. We report in Code 11.1 an example of *Level 2* configuration file.

Code 11.1 – Configuration file for *Level 2*.

```

1  directory=e2b_database/
2  npdir=false
3
4  [exporter]
5  element-spaces=P0
6
7  [eye2brain.patient]
8  name=tony
9  gender=0
10 age=42
11 note="Healthy subject - baseline"
12
13 [eye2brain.patient.assessment]
14 level=2
15 # circuit data
16 date=now
17 note="Healthy subject - baseline"
18 SBP=116
19 DBP=69
20 HR=69
21 IOP=17.0
22 RLTP=9.5
23 # Poisson data
24 CRVp=19.0
25 # Elasticity data
26 CRAp=80
27
28 [eye2brain]
29 # Poisson data
30 CRVp_marker=Hole
31 # Elasticity data
32 CRAp_marker=Hole
33 IOP_marker=Lamina_Retina
34 RLTP_marker=Out
35
36 [fmu]
37 filename=$cfgdir/../../eye2brain/models/fluid/retina/retina.fmu
38 solver.time-step=0.0002
39 #blood pressure drop on the lamina
40 exported-variables=lcR.v
41 # blood flow within the lamina
42 exported-variables=lcR.p.i
43 # CRA blood flow

```

```

44 exported-variables=R1a.p.i
45 # CRV blood flow
46 exported-variables=R5a.n.i
47
48 [gms]
49 filename=$cfgdir/eye.msh
50 submesh=Lamina
51 scale=0.1
52
53 [hdg.poisson]
54 filename=$cfgdir/lcDarcy.json
55
56 [hdg.elasticity]
57 gms.submesh=Lamina
58 filename=$cfgdir/lcElasticity.json
59
60 [bdf]
61 order=1
62 [ts]
63 time-initial=0.0
64 time-step=0.02
65 time-final=4
66 steady=false
67
68 [poroelastic]
69 itmax=2

```

In particular, the code shows how to specify

- the data of the patient: lines 7 – 11;
- the data of the assessment: lines 13 – 26;
- the information on the mesh: lines 28 – 34 and 48 – 51;
- the information on the 0D circuit: lines 36 – 46;
- the file where to retrieve the data for Darcy: line 54;
- the file where to retrieve the data for linear elasticity: line 58;
- the time simulation data: lines 60 – 66.

1.3 Darcy json file

We discuss now the file where we define all the data concerning the Darcy equations that are needed when using the `MixedPoisson` or `MixedPoissonElasticity` class. The type of this file is `json`, which is a file that stores simple data structures and objects in JavaScript Object Notation (JSON) format that is a standard data interchange format.

An example of Darcy `json` file is shown in Code 11.2, which can be easily adapted to any application.

Code 11.2 – `json` file for Darcy.

```

1 {
2   "Name": "HDG-Mixed-Poisson for lamina cribrosa",
3   "ShortName": "MP_LC",
4   "Models": { "equations": "hdg" },
5   "Materials":
6     {
7     "Lamina":
8       {
9         "name": "tissue",
10        "k": "0.015192",
11        "Lamina_Sclera-buffer": "C5.p.v",
12        "Cbuffer_name": "C5",
13        "Rbuffer_name": "lcRin"
14      }
9   }

```

```

15 },
16 "BoundaryConditions":
17 {
18   "potential":
19   {
20     "InitialSolution":
21     {
22       "Lamina":
23       {
24         "expr": "0.255"
25       }
26     },
27     "SourceTerm":
28     {
29       "Lamina":
30       {
31         "expr": "0.0"
32       }
33     },
34     "Dirichlet":
35     {
36       "Hole":
37       {
38         "expr": "0.25"
39       }
40     },
41     "Neumann":
42     {
43       "Lamina_Retina":
44       {
45         "expr": "0.0"
46       },
47       "Out":
48       {
49         "expr": "0.0"
50       }
51     }
52   },
53   "flux":
54   {
55     "Integral_coupled_with_0d":
56     {
57       "Lamina_Sclera":
58       {
59         "expr": "0.0"
60       }
61     }
62   }
63 },
64 "PostProcess":
65 {
66   "Exports":
67   {
68     "fields": ["potential", "flux"]
69   }
70 }
71 }

```

In this code we can find how to define

- the domain and its material properties: lines 9 – 10;
- the information on the coupling 3D-0D: lines 11 – 13;
- the initial solution $p(t = 0)$: lines 20 – 25;
- the volumetric source term f in Eq. (4.1b): lines 27 – 32;
- the Dirichlet BCs as Eq. (4.2): lines 34 – 40;
- the Neumann BCs as Eq. (4.3): lines 41 – 52;
- the integral interface condition as Eq. (7.4): lines 53 – 63;
- the quantities to be exported: lines 64 – 70.

1.4 Elasticity json file

Here we present the file where we define all the data concerning the linear elasticity equations that are needed when using the `MixedElasticity` or `MixedPoissonElasticity` class.

Code 11.3 – json file for linear elasticity.

```

1 {
2   "Name": "HDG-Mixed-Linear-Elasticity",
3   "ShortName": "ME",
4   "Models": { "equations": "hdg" },
5   "Materials":
6     {
7       "Lamina":
8         {
9           "name": "tissue",
10          "rho": "0.0011",
11          "lambda": "588",
12          "mu": "12",
13          "special_neumann": "Hole"
14        }
15      },
16   "BoundaryConditions":
17     {
18       "stress":
19         {
20           "SourceTerm":
21             {
22               "Lamina":
23                 {
24                   "expr": "(0,0,0)"
25                 }
26             },
27           "Neumann_scalar":
28             {
29               "Lamina_Retina":
30                 {
31                   "expr": "-0.33"
32                 },
33               "Out":
34                 {
35                   "expr": "-0.1"
36                 },
37               "Hole":
38                 {
39                   "expr": "-1.067"
40                 }
41             }
42         },
43       "displacement":
44         {
45           "Dirichlet":
46             {
47               "Lamina_Sclera":
48                 {
49                   "expr": "(0,0,0)"
50                 }
51             }
52         }
53     },
54   "PostProcess":
55     {
56       "Exports":
57         {
58           "fields": ["stress", "displacement"]
59         }
60     }
61 }

```

Similarly as for the other json file, we can identify in the code where we determine

- the domain and its material properties: lines 7 – 14; in particular lines 13 specifies the boundary for the poroelastic Neumann condition $\underline{\underline{\sigma}}_{el} \cdot \underline{\underline{n}} = -p_{cra} \underline{\underline{n}}$;
- the body force $\underline{\underline{F}}_{ext}$ of Eq. (5.1b): lines 20 – 26;
- the Neumann BCs as Eq. (5.2a): lines 27 – 41; in this particular case we define the variable g such that $\underline{\underline{g}}_N = g \underline{\underline{I}} \cdot \underline{\underline{n}}$;

- the Dirichlet BCs as Eq. (5.2b): lines 43 – 51;
- the quantities to be exported: lines 54 – 60.

All these input files can be easily modified with a text editor, however they require some familiarity in the usage of a Linux-base shell. For further details on how to run these applications we refer to the online documentation of FEEL++ available at <http://docs.feelpp.org/docs/stable/>.

1.5 Computational performances

As for the HDG toolboxes and the 3D-0D coupling implementation, we have completed a preliminary investigation on the computational time needed to solve the different levels of the OMVS.

- *Level 0* (Ch. 3) takes 6.17396 s to complete a simulation in serial with a final time of 4 s and a time step of 0.002 s.
- *Level 1* (Ch. 4) takes 4588.5 s (1.2746 hours) to complete a simulation using 24 processors with a final time of 4 s and a time step of 0.1 s. The dofs are 23484, 7828 and 12804 for the pressure, the flux and the trace, respectively.
- *Level 2a* (Ch. 5) takes 25144 s (6.984 hours) to complete a simulation using 24 processors with a final time of 4 s and a time step of 0.02 s. The dofs for the hemodynamics part are 578280, 192760 and 294948 for the pressure, the flux and the trace, respectively. The dofs for the biomechanics part are 1156560, 578280 and 897552 for the displacement, the stress and the trace, respectively.
- *Level 2b* (Ch. 5) takes 30842 s (8.567 hours) to complete a simulation using 24 processors with a final time of 2.3 s and a time step of 0.02 s. The dofs for the hemodynamics part are 134520, 44840 and 73281 for the pressure, the flux and the trace, respectively. The dofs for the biomechanics part are 2157168, 1078584 and 1707300 for the displacement, the stress and the trace, respectively.

These computational times are relative to the whole execution time of an OMVS application. For the hemodynamics part, thus the Darcy problem coupled with a system of ODEs, we employed the monolithic solver with LU as preconditioner [6]. For the biomechanics part, thus the linear elasticity equations, we exploit the static condensation technique with LU as preconditioner [6].

In the following we discuss the other interface, which is more user friendly and adapted to a clinical use.

2 OMVS user friendly interface

In view of real applications and in order to provide a clinical tool for the ophthalmologists, we developed a user friendly interface of the OMVS within the context of a European project

called MSO4SC (Mathematical Modeling, Simulation and Optimization for Societal Challenges with Scientific Computing). The MSO4SC proposal is to provide mathematical technologies and its applications for solution of societal challenges as service through an HPC oriented cloud e-infrastructure. In particular our model has been selected to be part of this project as the *Eye2Brain pilot*. The idea was to develop an online platform capable of providing clinicians with an integrated view of the patient's status, which would significantly improve our current ability to monitor health and prevent, detect, treat and manage disease in a personalized manner.

In this context we have theorized the concept of *virtual laboratory*, where ophthalmologists may perform clinical tests to improve the understanding, the diagnosis and the care of ocular diseases (*e.g.* glaucoma, diabetic retinopathy, age-related macular degeneration) and other pathologies that also manifest in the eye (*e.g.* diabetes, hypertension, NDD).

In this section we discuss the OMVS web interface, composed by three parts:

- the simulation interface;
- the View App;
- the Compare App.

2.1 OMVS simulation interface

The first overview of the MSO4SC portal is shown in Fig. 11.2. For the simulation of the Eye2Brain model, the user should select experiment and obtain a web-page similar to the one illustrated by Fig. 11.3.

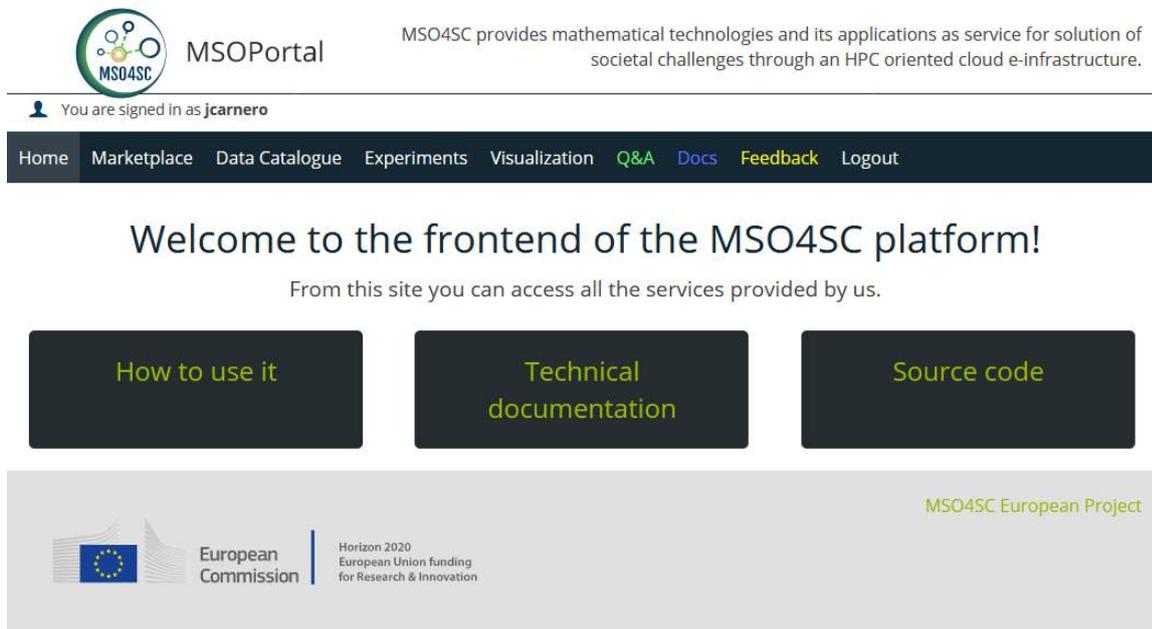
Thus, we create a new instance, which corresponds to our patient, selecting the Eye2Brain model we would like to use:

- (i) `eye2brain-level-0`, which corresponds to the model described in Ch. 3;
- (ii) `eye2brain-level-1`, which corresponds to the model described in Ch. 4;
- (iii) `eye2brain-level-2`, which corresponds to the models described in Ch. 5.

The choice between the different levels depends on the accuracy and speed needed by the clinical application. For all selections, the input are already filled with some default values in order to have always all the parameters specified. The input parameter interfaces are displayed in Figs. 11.4, 11.5, and 11.6 for `eye2brain-level-0`, `eye2brain-level-1`, and `eye2brain-level-2`, respectively.

For seek of clarity we report the input that can be modified by the user in Tabs. 11.3, 11.4, and 11.5.

Finally, the models are solved using the advanced computational methods described in Part III in external HPC facilities and the simulation results saved on private online repositories.



MSO4SC provides mathematical technologies and its applications as service for solution of societal challenges through an HPC oriented cloud e-infrastructure.

You are signed in as **jcarnero**

Home Marketplace Data Catalogue Experiments Visualization Q&A Docs Feedback Logout

Welcome to the frontend of the MSO4SC platform!

From this site you can access all the services provided by us.

[How to use it](#) [Technical documentation](#) [Source code](#)

MSO4SC European Project

European Commission | Horizon 2020 European Union funding for Research & Innovation

Figure 11.2 – Overview of the MSO4SC portal.

Key name	Default value	Unit	Brief description
DBP	69	mmHg	Diastolic Blood Pressure
SBP	116	mmHg	Systolic Blood Pressure
HR	69	bpm	Heart Rate
IOP	17	mmHg	IntraOcular Pressure
RLTp	9.5	mmHg	RetroLaminar Tissue Pressure

Table 11.3 – Input parameters for `eye2brain-level-0`.

Key name	Default value	Unit	Brief description
DBP	69	mmHg	Diastolic Blood Pressure
SBP	116	mmHg	Systolic Blood Pressure
HR	69	bpm	Heart Rate
IOP	17	mmHg	IntraOcular Pressure
RLTp	9.5	mmHg	RetroLaminar Tissue Pressure
CRVp	19	mmHg	Central Retinal Vein Pressure
κ	0.015192	$\text{cm}^3 \text{s Kg}^{-1}$	Permeability coefficient

Table 11.4 – Input parameters for `eye2brain-level-1`.

Home Marketplace Data Catalogue Experiments Visualization Q&A Docs Feedback Logout

Instances Settings

→ Add a new one

Application:
None

Deployment ID:

Create new Instance

App's instances:
No instances found

Run! Force Destroy Instance! Destroy Instance!

Figure 11.3 – Experiments in the MSO4SC portal.

Key name	Default value	Unit	Brief description
DBP	69	mmHg	Diastolic Blood Pressure
SBP	116	mmHg	Systolic Blood Pressure
HR	69	bpm	Heart Rate
IOP	17	mmHg	IntraOcular Pressure
RLTp	9.5	mmHg	RetroLaminar Tissue Pressure
CRVp	19	mmHg	Central Retinal Vein Pressure
κ	0.015192	$\text{cm}^3 \text{ s Kg}^{-1}$	Permeability coefficient
ρ	0.0011	Kg cm^{-3}	Tissue density
λ	5.88	GPa	Lamé first parameter
μ	0.12	GPa	Lamé second parameter

Table 11.5 – Input parameters for eye2brain-level-2.

Applications Instances Settings

→ Add a new one

Application:
eye2brain-level-0-minimal

Deployment ID:
eye-l1-minimal

e2b_patient_11_ID
e2b-patient

e2b_patient_12_gender
0

e2b_patient_13_age
40

e2b_patient_14_Note
Patient note

e2b_patient_2_Date
now

e2b_patient_3_DBP
69

e2b_patient_4_SBP
116

e2b_patient_5_HR
69.0

e2b_patient_6_IOP
17.0

e2b_patient_7_RLTP
9.5

e2b_patient_99_Note
Patient assessment note

HPC: primary
FT6

Figure 11.4 – Input parameter interface for eye2brain-level-0

Application:
eye2brain-level-1-minimal

Deployment ID:
eye-l1-minimal

e2b_patient_11_ID
e2b-patient

e2b_patient_12_gender
0

e2b_patient_13_age
40

e2b_patient_14_Note
Patient note

e2b_patient_2_Date
now

e2b_patient_3_DBP
69

e2b_patient_4_SBP
116

e2b_patient_5_HR
69.0

e2b_patient_6_IOP
17.0

e2b_patient_7_RLTP
9.5

e2b_patient_80_CRVp
19

e2b_patient_99_Note
Patient assessment note

HPC: primary
FT6

Create new Instance

Figure 11.5 – Input parameter interface for eye2brain-level-1

Application:
eye2brain-level-2-minimal

Deployment ID:

e2b_patient_11_ID
e2b-patient

e2b_patient_12_gender
0

e2b_patient_13_age
40

e2b_patient_14_Note
Patient note

e2b_patient_2_Date
now

e2b_patient_3_DBP
69

e2b_patient_4_SBP
116

e2b_patient_5_HR
69.0

e2b_patient_6_IOP
17.0

e2b_patient_7_RLTp
9.5

e2b_patient_80_CRVp
19

e2b_patient_90_CRAp
80

e2b_patient_99_Note
Patient assessment note

HPC: primary
FT6

Figure 11.6 – Input parameter interface for eye2brain-level-2

2.2 OMVS View App

To visualize the simulation results that are stored in a private repository, we have implemented two online Apps, the first one is called *OMVS View App*. The idea is to select the patient from the online database that is automatically update from the simulation interface, and then visualize the computed results.

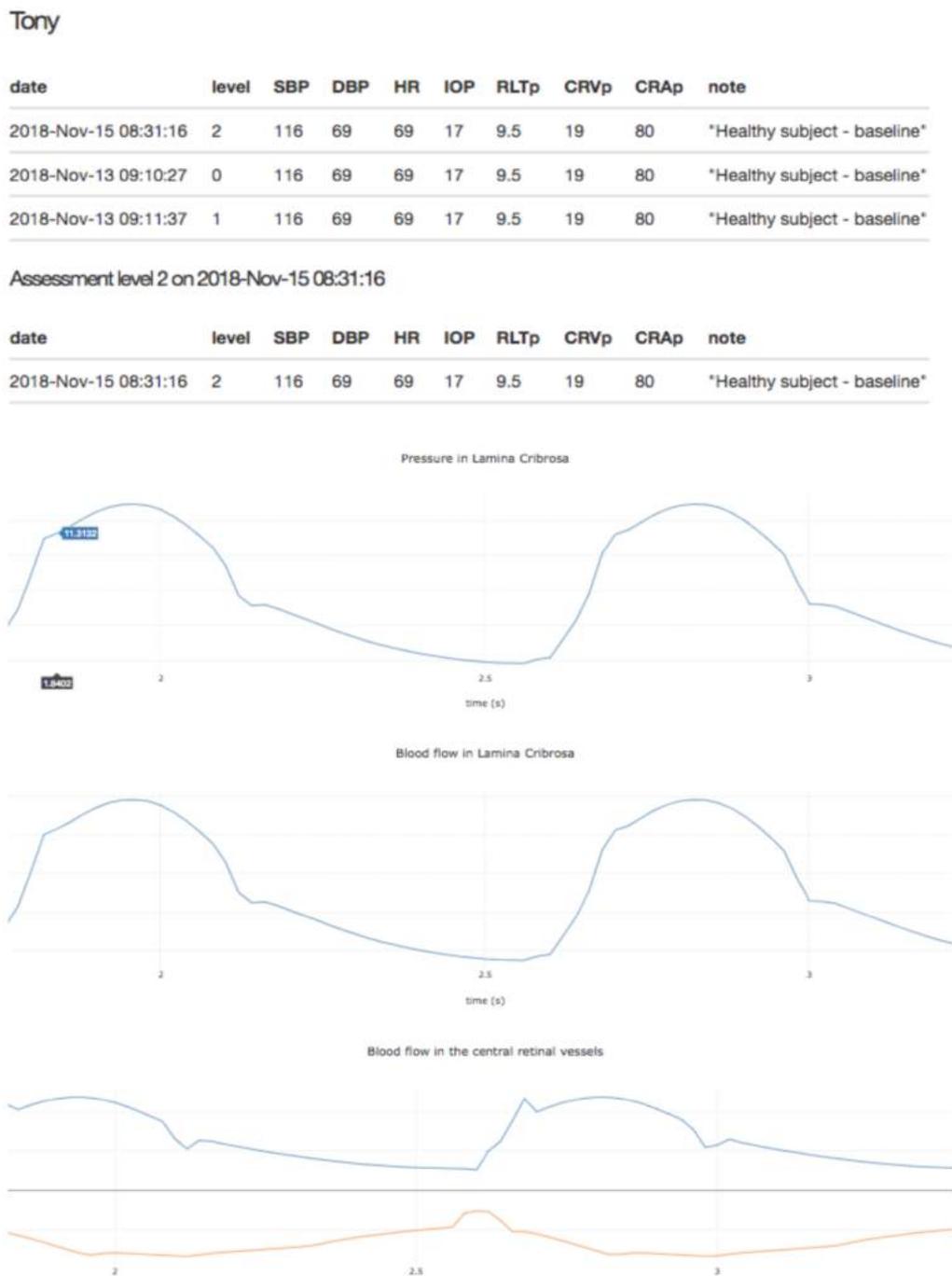


Figure 11.7 – OMVS View App interface.

Fig. 11.7 shows the current interface of the OMVS View App; in particular we have a first

part where the online platform allows the user to select the desired patient and recap the input data used in the simulation. Below the input data table, we report the output data that are currently available:

- average blood pressure profile in the lamina cribrosa;
- average blood flow profile in the lamina cribrosa;
- average blood flow profile in the central retinal vessels.

This last outcome is very similar to the output given by the Color Doppler Imaging [102].

2.3 OMVS Compare App

The OMVS Compare App works similarly to the View App, however in this case the user can select two different patients or two different simulations of the same patients. Ideally the comparison between two patients would serve as virtual clinical laboratory to understand the effects of different set of input on relevant output, whereas the comparison between two simulation results of the same patient would reproduce the situation of follow-up of a patient during a treatment, where the ophthalmologist can examine the evolution of the disease in the patient.

In Fig. 11.8 we display the interface of the OMVS Compare App. We resume the input data of the two sets of the results that need to be compared in two tables, and below we report the computed results. In particular the outcome that are currently available on the online platform are:

- average blood pressure profile in the lamina cribrosa;
- average blood flow profile in the lamina cribrosa;
- average blood flow profile in the central retinal vessels.

All the other output computed by our model are not yet available on the online user friendly interface, however the result files are accessible directly in a private repository on Girder^{*}. Girder is a free and open source web-based data management platform developed by Kitware[†] for data organization and dissemination, user management and authentication, and authorization management.

More details and a tutorial on the usage of this OMVS interface are available at <http://docs.cemosis.fr/eye2brain/current/interface/>.

^{*}<https://girder.readthedocs.io/en/stable/>

[†]<https://www.kitware.com/>

Tony vs John

date	level	SBP	DBP	HR	IOP	RLTp	CRVp	CRAp	note
2018-Nov-13 09:10:27	0	116	69	69	17	9.5	19	80	"Healthy subject - baseline"
2018-Nov-13 09:10:41	0	116	69	69	26	9.5	19	80	"High IOP patient - glaucoma risk"

Patient1 vs Patient2

date	level	SBP	DBP	HR	IOP	RLTp	CRVp	CRAp	note
2018-Nov-13 09:10:27	0	116	69	69	17	9.5	19	80	"Healthy subject - baseline"
2018-Nov-13 09:10:41	0	116	69	69	26	9.5	19	80	"High IOP patient - glaucoma risk"

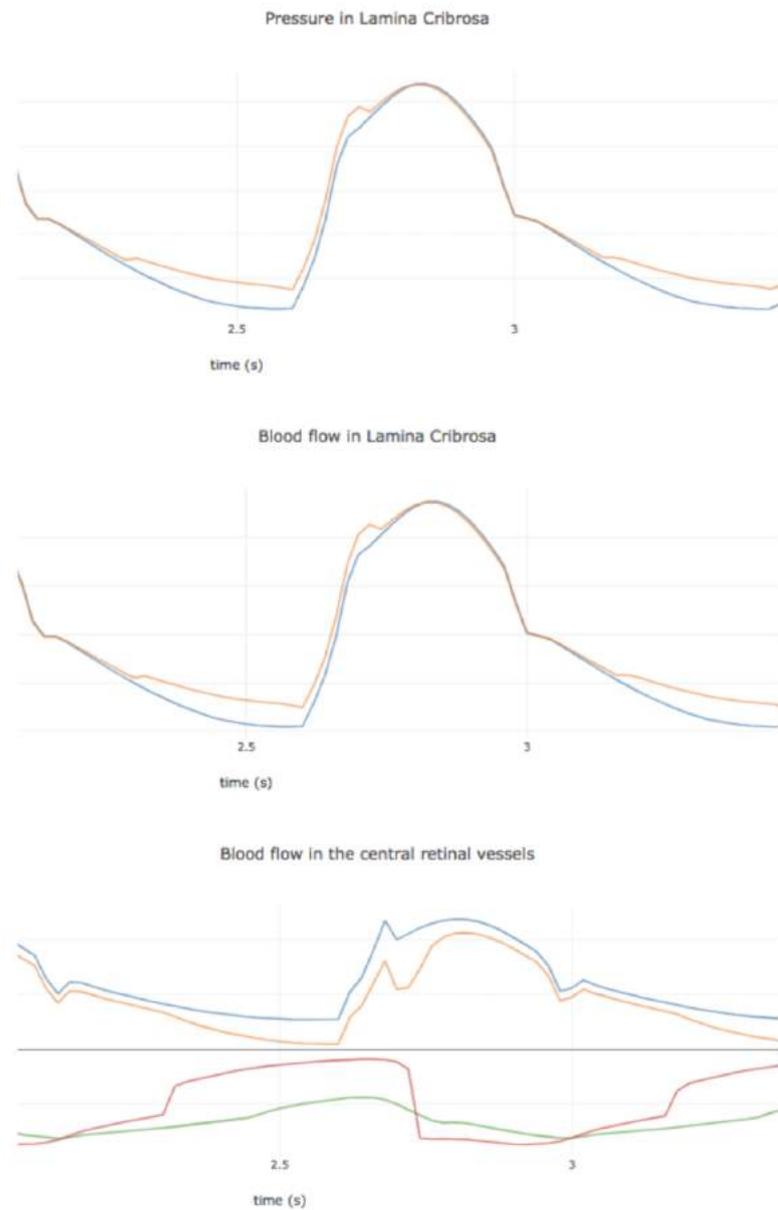


Figure 11.8 – OMVS Compare App interface.

Part V

Simulation results

In this part we display the simulation results provided by the OMVS. In particular we present:

- the verification of the implementation (Part IV) of the numerical methods introduced in Part III (Ch. 12);
- the validation of the models described in Part II using clinical or experimental data (Ch. 13);
- three clinically significant studies to investigate different aspects of the ocular physiology, namely the effects on the overall system caused by a variation of the blood pressure (Ch. 14), of the intraocular pressure (Ch. 15) and of the translaminal tissue pressure difference (Ch. 16);
- the application of our framework to a clinical virtual database (Ch. 17).

Contents

1	HDG convergence study	133
1.1	Darcy convergence test	134
1.2	Linear elasticity convergence test	137
2	Operator splitting convergence study	145
2.1	Linear test case	147
2.2	Nonlinear resistor test case	151
2.3	Nonlinear resistor and capacitor test case	154

In this chapter we provide the verification of the numerical methods presented in chapters 7 and 8. In the first section we illustrate the convergence study for the HDG method applied for solving Darcy equations and the linear elasticity system, while in the second section we discuss the convergence study for the proposed operator splitting for the coupled Darcy-ODE system.

1 HDG convergence study

In this section we report the convergence results of the HDG algorithm for the Darcy equations and for the linear elasticity problem (Ch. 8) using the implementation in FEEL++ illustrated in chapter 9.

We have performed the convergence tests in 2D and in 3D using the same computational domain for Darcy and linear elasticity model (Figs. 12.1). Notice that the 3D domain is obtained by extruding the 2D one.

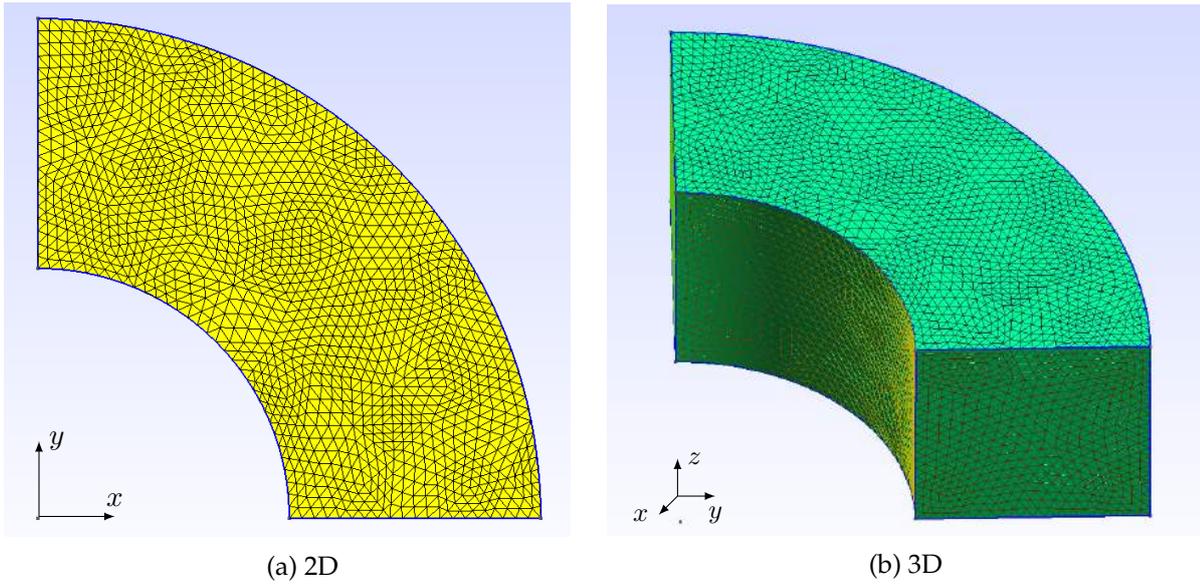


Figure 12.1 – Computational domains used for the convergence test.

Let us denote by Ω the 2D/3D domain and $\Gamma = \partial\Omega$. For the sake of clarity we define the boundaries of the 2D computational domain displayed in Fig. 12.1a:

$$\Gamma_1 = \{(x, y) \in \Gamma : y = 0\} \quad (12.1a)$$

$$\Gamma_2 = \{(x, y) \in \Gamma : x = 0\} \quad (12.1b)$$

$$\Gamma_3 = \Gamma \setminus (\Gamma_1 \cup \Gamma_2) \quad (12.1c)$$

and the 3D boundaries showed in Fig. 12.1b:

$$\Gamma_1 = \{(x, y, z) \in \Gamma : y = 0\} \quad (12.2a)$$

$$\Gamma_2 = \{(x, y, z) \in \Gamma : x = 0\} \quad (12.2b)$$

$$\Gamma_3 = \{(x, y, z) \in \Gamma : z = 0\} \cup \{(x, y, z) \in \Gamma : z = H\} \quad (12.2c)$$

$$\Gamma_4 = \Gamma \setminus (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3) \quad (12.2d)$$

where H is the dimension of the extrusion along z of the 3D domain with respect to the bi-dimensional one.

1.1 Darcy convergence test

In Eq. (4.1), we assume that $\underline{\mathcal{K}}$ is the identity matrix. The analytical solutions that we use for the Darcy convergence test read:

$$\underline{j}(x, y) = \frac{1}{2\pi} \begin{pmatrix} \frac{-y}{x^2+y^2} \\ \frac{x}{x^2+y^2} \end{pmatrix}, \quad p(x, y) = \frac{1}{2\pi} \text{atan2}(y, x) \quad \text{in } \Omega,$$

in 2D (Fig. 12.2), and

$$\underline{j}(x, y, z) = \frac{\text{atan2}(y, x) \cos(xyz)}{2\pi} \begin{pmatrix} yz \\ xz \\ xy \end{pmatrix} + \frac{1 + \sin(xyz)}{2\pi(x^2 + y^2)} \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix} \quad \text{in } \Omega,$$

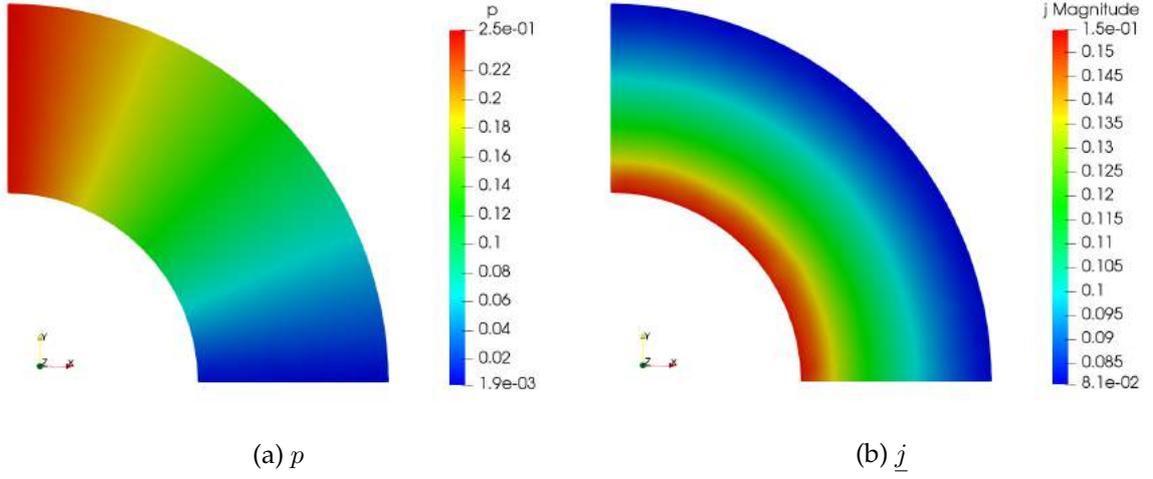


Figure 12.2 – Analytical solutions in 2D.

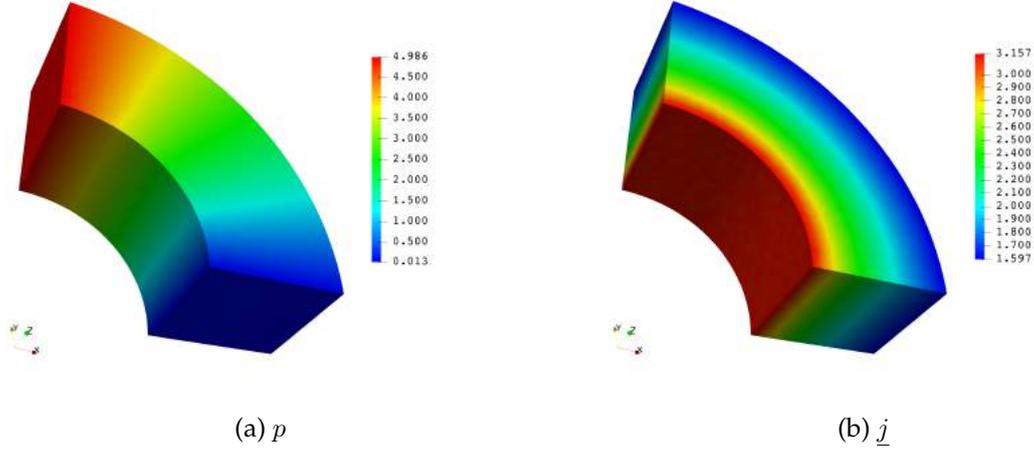


Figure 12.3 – Analytical solutions in 3D.

$$p(x, y, z) = \frac{1}{2\pi} \text{atan2}(y, x)(1 + \sin(xyz)) \quad \text{in } \Omega$$

in 3D (Fig. 12.3).

We perform two different sets of simulations, a first set without any IBC condition (*w/o IBC*), *i.e.* when only standard Dirichlet and Neumann boundary conditions are applied, then a second one with IBC (*w/ IBC*).

w/o IBC

Considering the 2D domain sketched in Fig. 12.1a, we define the boundaries as follow for the *w/o IBC* test:

$$\Gamma_D = \Gamma_1 \cup \Gamma_2, \quad \Gamma_N = \Gamma_3.$$

For the 3D-domain (Fig. 12.1b), we have

$$\Gamma_D = \Gamma_1 \cup \Gamma_2,$$

$$\Gamma_N = \Gamma_3 \cup \Gamma_4.$$

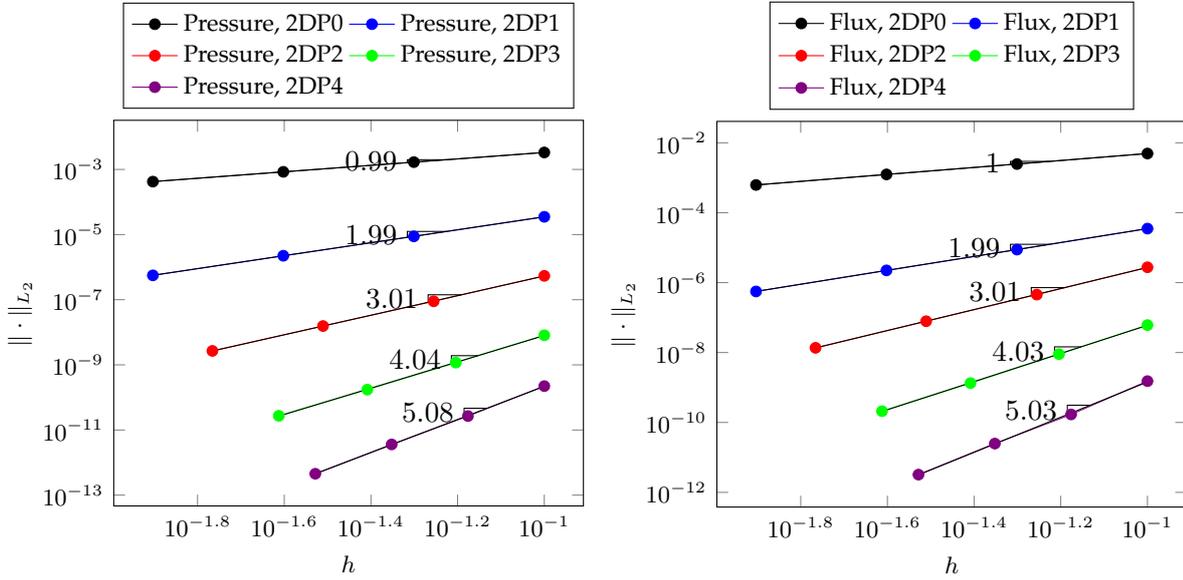


Figure 12.4 – Numerical convergence order for the 2D Darcy *w/o* IBC test. Left panel: pressure p convergence study. Right panel: flux j convergence study.

Fig. 12.4 shows the computed convergence order for the 2D *w/o* IBC test. The simulated orders are in agreement with the theoretical ones using polynomials (finite element spaces P^k) up to fourth order.

We obtain similar outcomes for the 3D *w/o* IBC study displayed in Fig. 12.5.

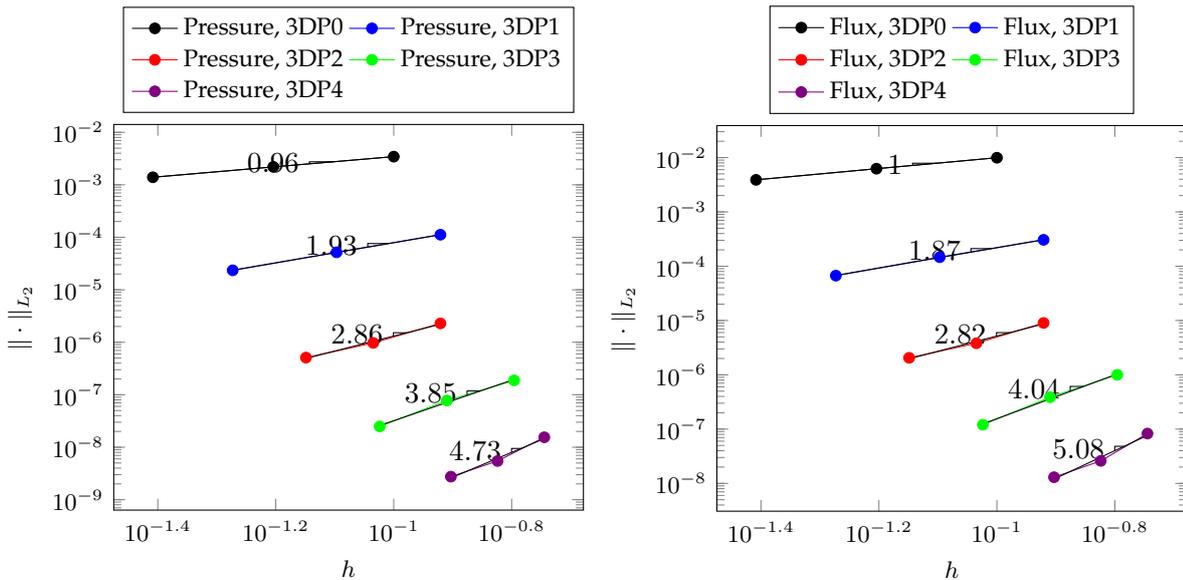


Figure 12.5 – Numerical convergence order for the 3D Darcy *w/o* IBC test. Left panel: pressure p convergence study. Right panel: flux j convergence study.

w/ IBC

Considering the 2D domain sketched in Fig. 12.1a, we define the boundaries as follows for the w/IBC test case:

$$\Gamma_D = \Gamma_1, \quad \Gamma_I = \Gamma_2, \quad \Gamma_N = \Gamma_3.$$

For the 3D-domain (Fig. 12.1b), we have

$$\Gamma_D = \Gamma_1, \quad \Gamma_I = \Gamma_2, \quad \Gamma_N = \Gamma_3 \cup \Gamma_4.$$

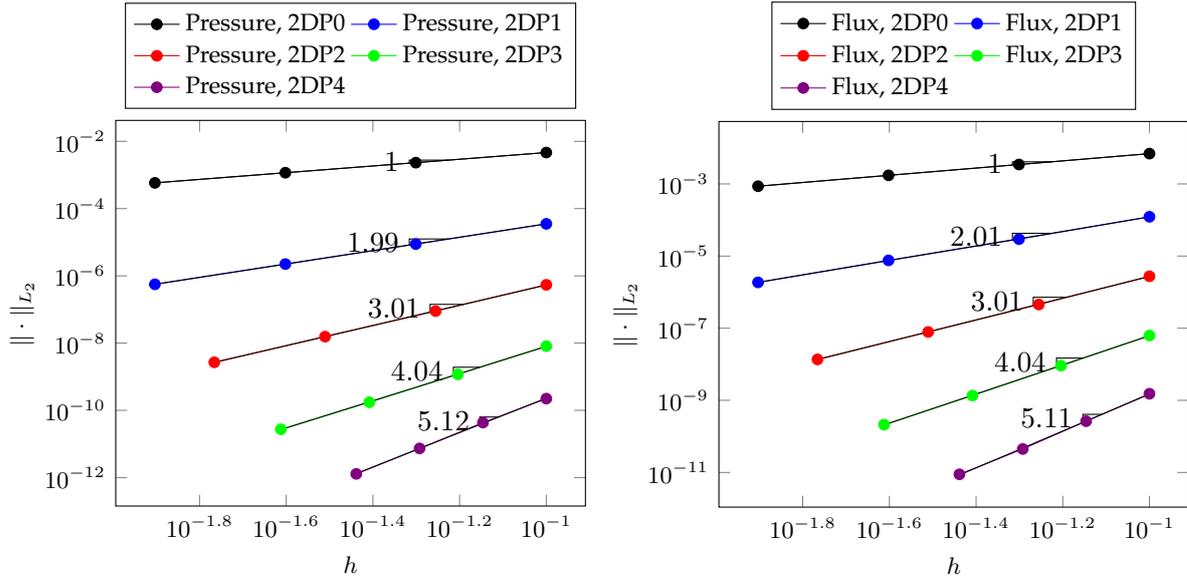


Figure 12.6 – Numerical convergence order for the 2D Darcy w/IBC test. Left panel: pressure p convergence study. Right panel: flux \underline{j} convergence study.

Figs. 12.6 and 12.7 show the computed convergence orders for Darcy problems with IBC in 2D and 3D, respectively. Also in this case we achieve the correct orders predicted by the theory [43].

In Tab. 12.1 we resume the computed orders for the 2D and 3D study, $w/$ and w/o IBC, up to fourth polynomial order. These computed orders are in agreement with the theoretical predictions, with the exception of the 3D third polynomial order w/IBC case. We remind, however, that we haven't used the L_2 -projector P_M , therefore we may not obtain the optimal convergence (see Remark 10 in Ch. 8).

1.2 Linear elasticity convergence test

For all the linear elasticity convergence tests we have used the two unitary Lamé parameters, namely $\lambda = 1$ and $\mu = 1$.

2D

We simulate three different situations, called *Dirichlet*, *Neumann*, and *IBC*. As the names suggest, we have used different type of boundary conditions, which are resumed in Tab. 12.2.

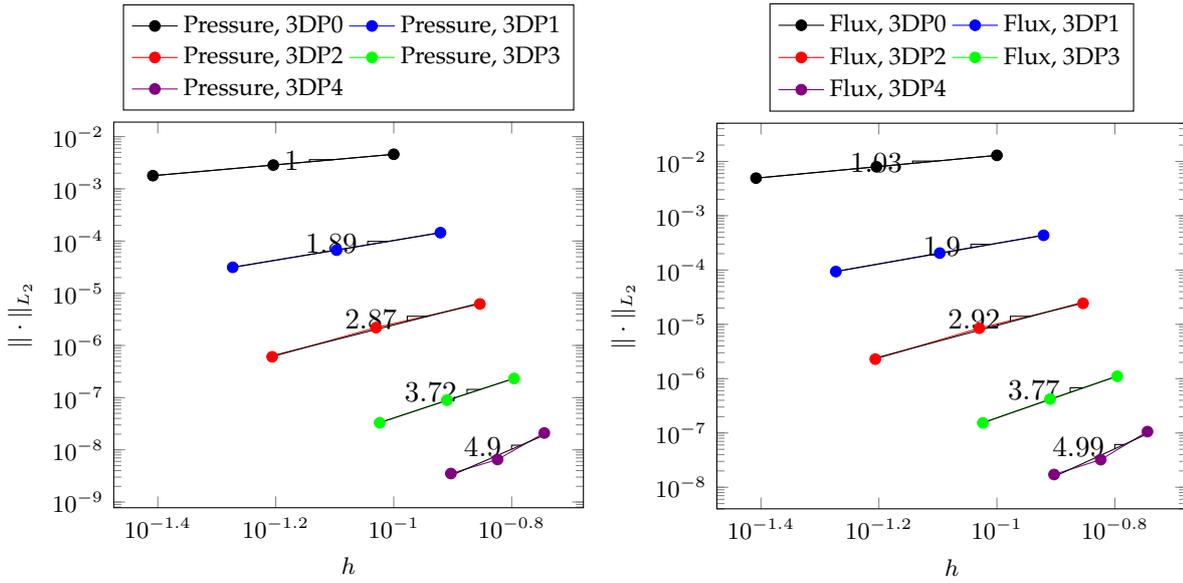


Figure 12.7 – Numerical convergence order for the 3D Darcy w/IBC test. Left panel: pressure p convergence study. Right panel: flux j convergence study.

		2D									
		P0		P1		P2		P3		P4	
		p	j	p	j	p	j	p	j	p	j
w/o	IBC	0.99	1	1.99	1.99	3.01	3.01	4.04	4.03	5.08	5.03
$w/$	IBC	1	1	1.99	2.01	3.01	3.01	4.04	4.04	5.12	5.11

		3D									
		P0		P1		P2		P3		P4	
		p	j								
w/o	IBC	0.96	1	1.93	1.87	2.86	2.82	3.85	4.04	4.73	5.08
$w/$	IBC	1	1.03	1.89	1.9	2.87	2.92	3.72	3.77	4.9	4.99

Table 12.1 – Numerical orders for the Darcy convergence study.

For what concerns the analytical solutions, we have used the following expressions:

- *Dirichlet*

$$\underline{u}_{exact}(x, y) = \left[\frac{1}{2\pi^2} \sin(\pi x) \cos(\pi y), \frac{1}{2\pi^2} \cos(\pi x) \sin(\pi y) \right]$$

$$\underline{F}_{ext,exact}(x, y) = [-3 \sin(\pi x) \cos(\pi y), -3 \cos(\pi x) \sin(\pi y)]$$

- *Neumann*

$$\underline{u}_{exact}(x, y) = \left[\frac{1}{2\pi^2} \sin(\pi x) \cos(\pi y), \frac{1}{2\pi^2} \cos(\pi x) \sin(\pi y) \right]$$

$$\underline{F}_{ext,exact}(x, y) = [-3 \sin(\pi x) \cos(\pi y), -3 \cos(\pi x) \sin(\pi y)]$$

- *IBC*

$$\underline{u}_{exact}(x, y) = [\sin(\pi x) \sin(\pi y), \cos(\pi x) \sin(\pi y)]$$

Name test	$\Gamma_{Dirichlet}$	$\Gamma_{Neumann}$	Γ_{IBC}
Dirichlet	$\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$	\emptyset	\emptyset
Neumann	$\Gamma_1 \cup \Gamma_2$	Γ_3	\emptyset
IBC	Γ_2	Γ_3	Γ_1

Table 12.2 – Boundary conditions for 2D convergence tests

$$\underline{F}_{ext,exact}(x, y) = [-2\pi^2 \sin(\pi x) (2 \sin(\pi y) + \cos(\pi y)) , 2\pi^2 \cos(\pi x) (\cos(\pi y) - 2 \sin(\pi y))]$$

$$\int_{\Gamma_{IBC}} \underline{\sigma}_{exact}(x, y) \cdot \underline{n} = [2, 0]$$

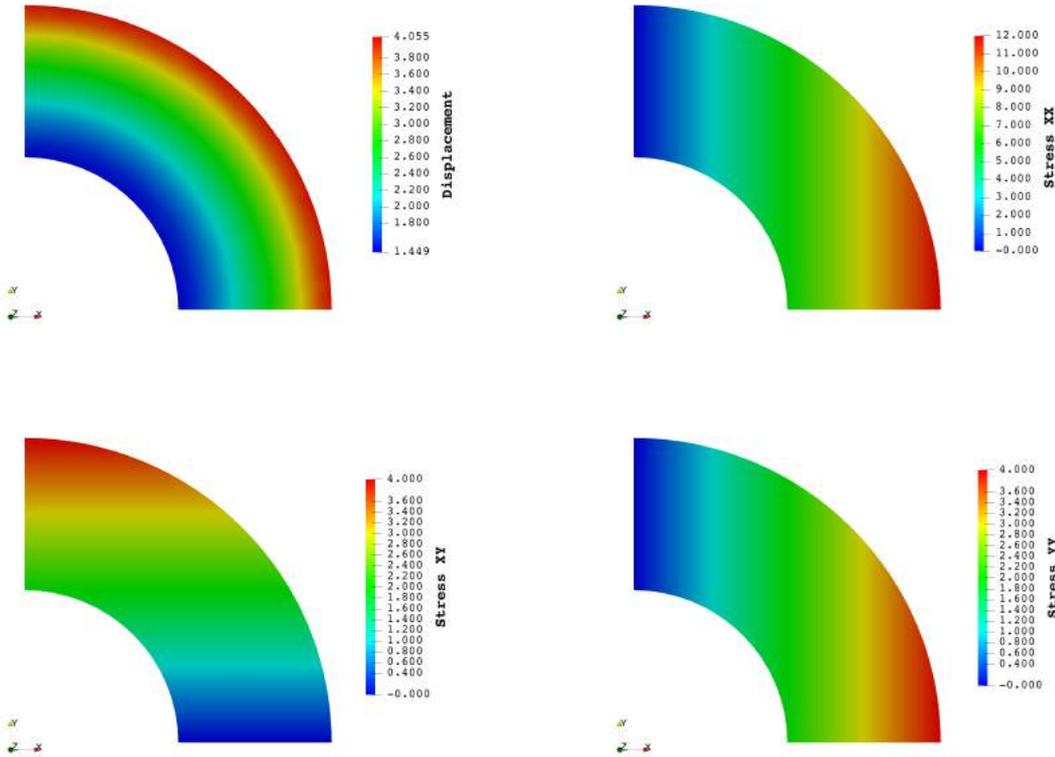


Figure 12.8 – Visualization of the 2D analytical solution on the computational mesh.

Fig. 12.8 illustrates the spatial distribution of the exact solution for the 2D convergence tests. The simulated convergence orders for the three different tests are computed and shown by Figs. 12.9, 12.10 and 12.11 up to the fourth polynomial order.

The 2D results and most of the computed convergence orders (Tab. 12.3) provided by FEEL++ agree with the theoretical predictions presented in Ch. 8. We remind that we haven't used the L_2 -projector P_M , therefore we may not obtain the optimal convergence order for all cases (see Remark 10 in Ch. 8).

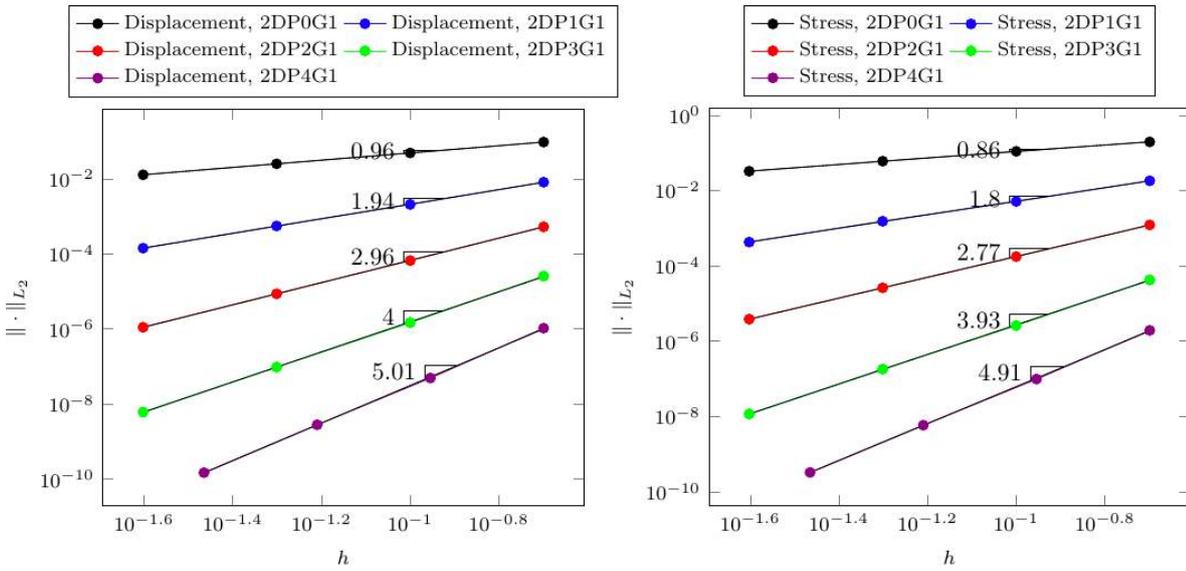


Figure 12.9 – Numerical convergence order for the 2D *Dirichlet* test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\sigma}$ convergence study.

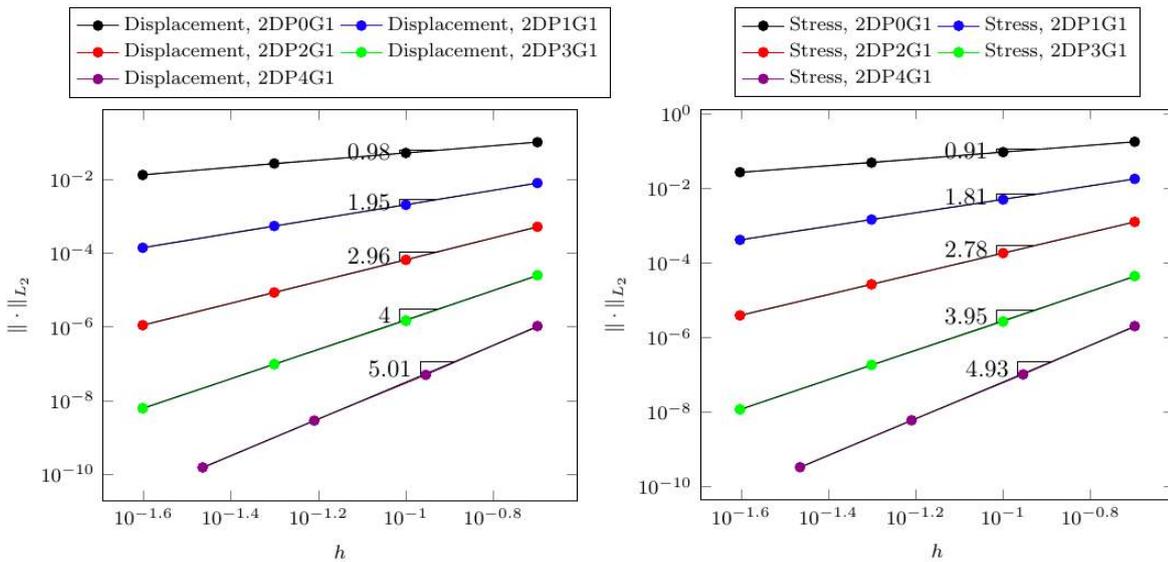


Figure 12.10 – Numerical convergence order for the 2D *Neumann* test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\sigma}$ convergence study.

3D

Also for 3D we simulate three different situations, *Dirichlet*, *Neumann*, and *IBC*. The specific details of the boundary conditions for these three tests are resumed in Tab. 12.4.

For the 3D convergence study we have used the same analytical solution (Fig. 12.12) for the three test cases:

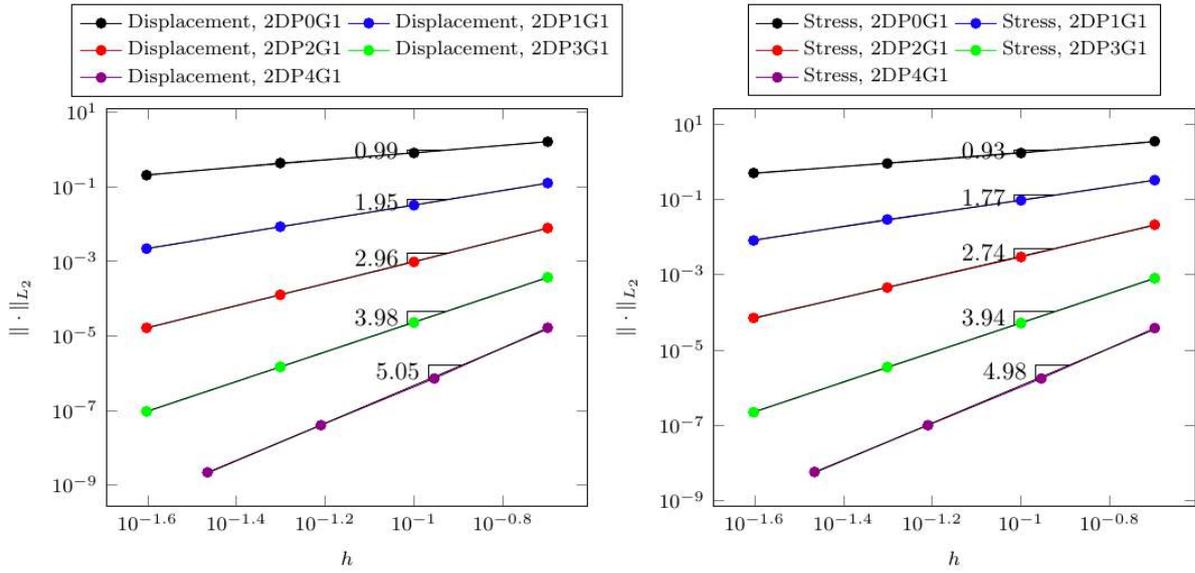


Figure 12.11 – Numerical convergence order for the 2D IBC test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\sigma}$ convergence study.

Variable	P ₀	P ₁	P ₂	P ₃	P ₄
DIRICHLET					
\underline{u}	0.96	1.94	2.96	4	5.01
$\underline{\sigma}$	0.86	1.8	2.77	3.93	4.91
NEUMANN					
\underline{u}	0.98	1.95	2.96	4	5.01
$\underline{\sigma}$	0.91	1.81	2.78	3.95	4.93
IBC					
\underline{u}	0.99	1.95	2.96	3.98	5.05
$\underline{\sigma}$	0.93	1.77	2.74	3.94	4.98

Table 12.3 – Numerical convergence order for 2D convergence study.

$$\underline{u}_{exact}(x, y, z) = [\cos(\pi x) \cos(\pi y) \cos(\pi z), \sin(\pi x) \cos(\pi y) \sin(\pi z), \cos(\pi x) \sin(\pi y) \cos(\pi z)]$$

$$\underline{F}_{ext,exact}(x, y, z) = \begin{bmatrix} 2\pi^2 \sin(\pi x) \sin(\pi y) \sin(\pi z) - 2\pi^2 \cos(\pi x) \sin(\pi y) \sin(\pi z) - 5\pi^2 \cos(\pi x) \cos(\pi y) \cos(\pi z) \\ 2\pi^2 \cos(\pi z) \sin(\pi x) \sin(\pi y) - 5\pi^2 \cos(\pi y) \sin(\pi x) \sin(\pi z) - 2\pi^2 \cos(\pi x) \cos(\pi y) \sin(\pi z) \\ 2\pi^2 \cos(\pi y) \sin(\pi x) \sin(\pi z) - 5\pi^2 \cos(\pi x) \cos(\pi z) \sin(\pi y) - 2\pi^2 \cos(\pi z) \sin(\pi x) \sin(\pi y) \end{bmatrix}$$

The numerical convergence orders for the three different tests in the 3D domain are computed and illustrated by Figs. 12.13, 12.14 and 12.15 up to the fourth polynomial order.

The 3D results and most of the numerical convergence orders (Tab. 12.5) provided by FEEL++ agree with the theoretical predictions presented in Ch. 8. We remind that we haven't used the L_2 -projector P_M , therefore we may not obtain the optimal convergence order for all cases (see Remark 10 in Ch. 8).

Name test	$\Gamma_{Dirichlet}$	$\Gamma_{Neumann}$	Γ_{IBC}
Dirichlet	$\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$	\emptyset	\emptyset
Neumann	$\Gamma_1 \cup \Gamma_2$	$\Gamma_3 \cup \Gamma_4$	\emptyset
IBC	Γ_2	$\Gamma_3 \cup \Gamma_4$	Γ_1

Table 12.4 – Boundary conditions for 3D convergence tests.

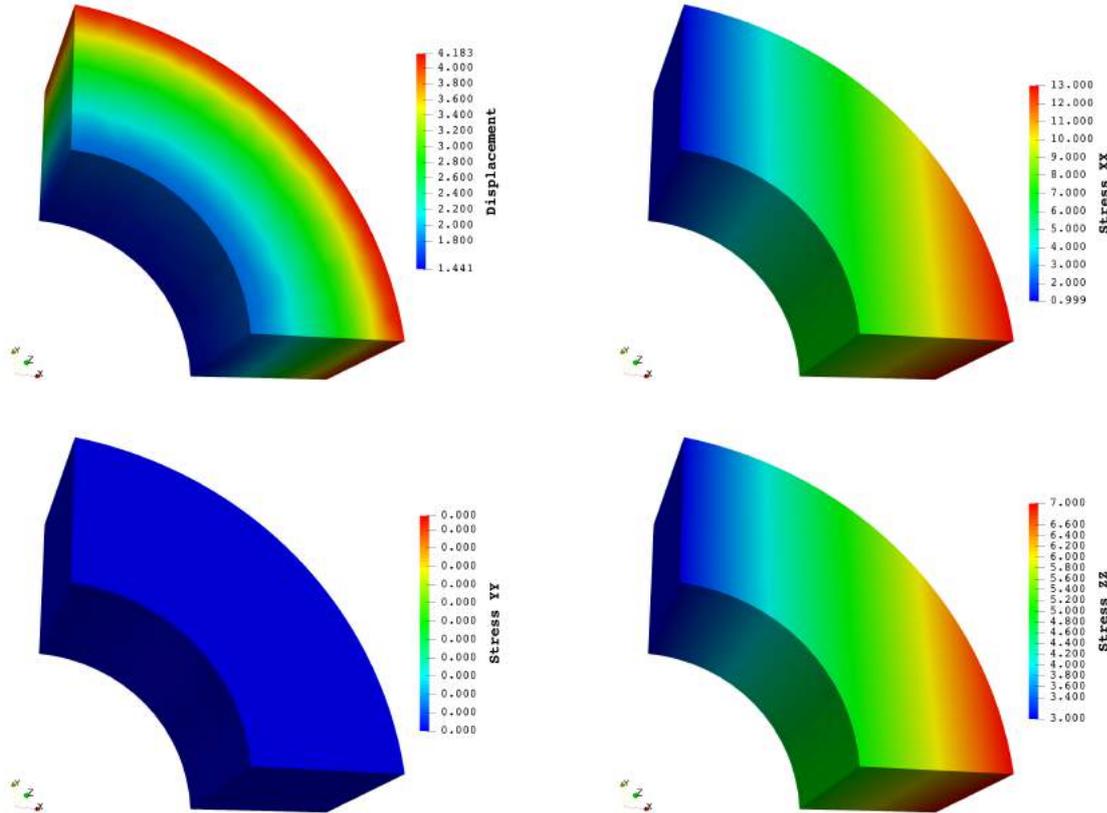


Figure 12.12 – Visualization of the 3D analytical solution on the computational mesh.

Variable	P_0	P_1	P_2	P_3	P_4
DIRICHLET					
\underline{u}	0.89	1.8	3.05	4.42	6.55
$\underline{\sigma}$	0.77	1.71	3	4.4	6.61
NEUMANN					
\underline{u}	1.01	2.02	3.06	4.42	5.65
$\underline{\sigma}$	0.83	1.91	3	4.39	5.74
IBC					
\underline{u}	1.04	2.03	3.08	4.4	5.46
$\underline{\sigma}$	0.87	1.92	3.03	4.42	5.62

Table 12.5 – Numerical convergence order for 3D convergence study.

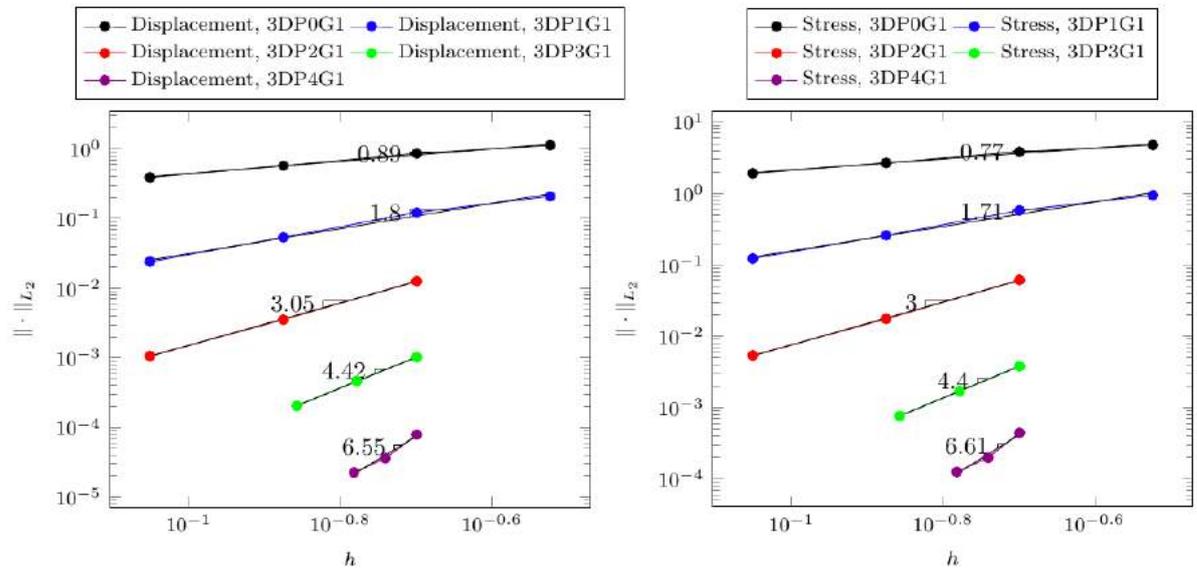


Figure 12.13 – Numerical convergence order for the 3D *Dirichlet* test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\sigma}$ convergence study.

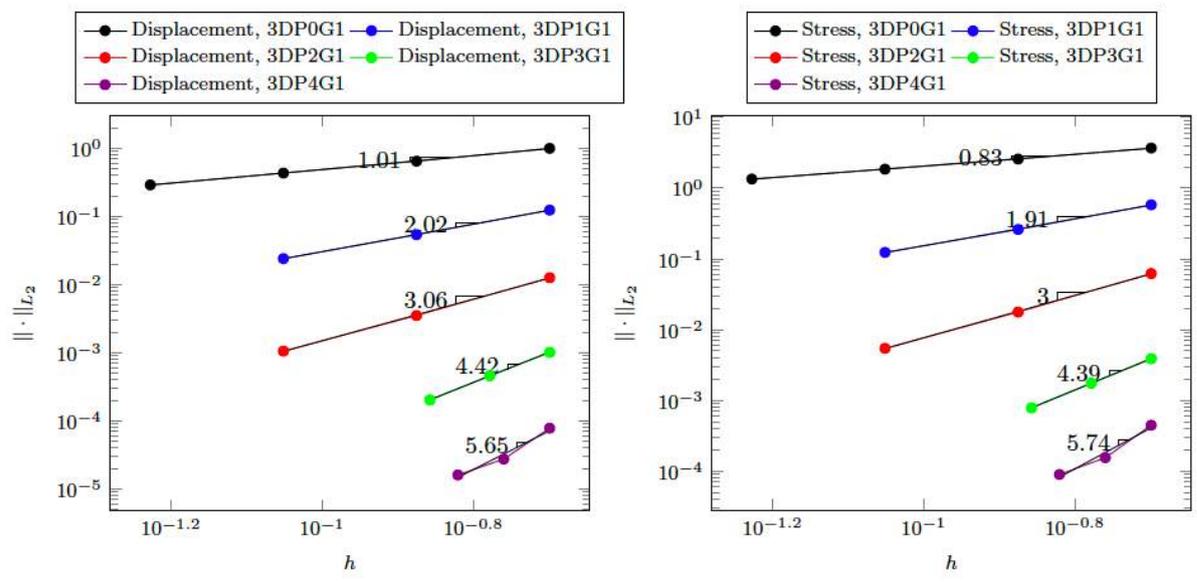


Figure 12.14 – Numerical convergence order for the 3D *Neumann* test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\sigma}$ convergence study.

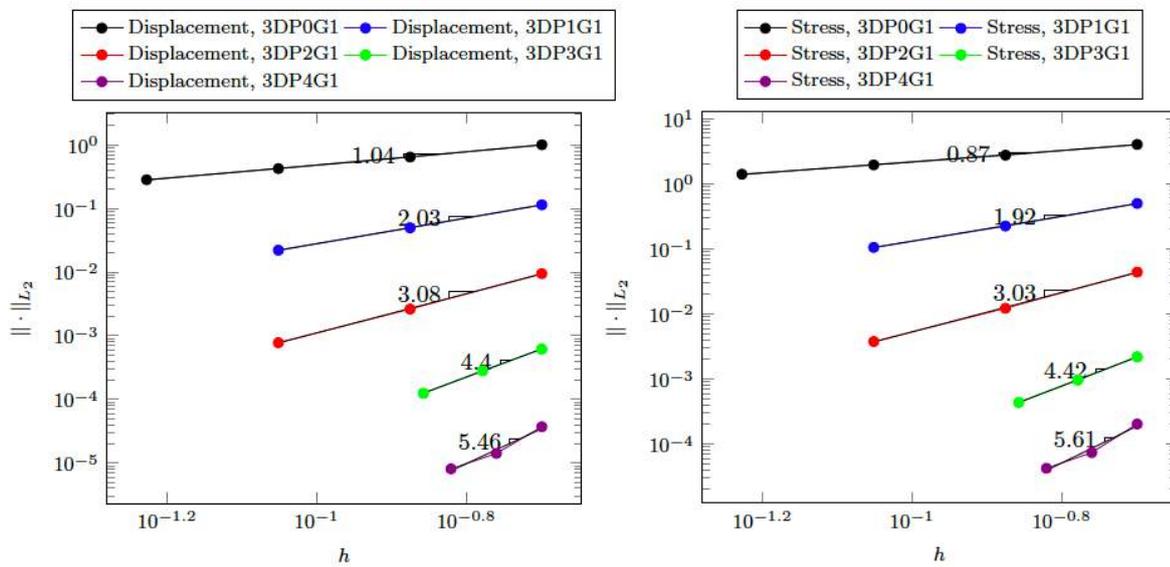


Figure 12.15 – Numerical convergence order for the 3D IBC test. Left panel: displacement \underline{u} convergence study. Right panel: stress $\underline{\underline{\sigma}}$ convergence study.

Geometric test

Finally we present a preliminary investigation of the impact of the geometric approximation, which we have performed on the 2D computational domain (Fig. 12.1a). In particular, for this geometric test, we have computed the force \tilde{F} on Γ_2 and compared the results obtained using first geometric order and second geometric order with respect to the analytical computation of this force.

The analytical expression used for this study are:

$$\begin{aligned} \underline{u}_{exact}(x, y) &= [x^5, y^5] \\ \underline{F}_{ext,exact}(x, y) &= [60x^3, 60y^3] \\ \tilde{F}_{exact}(x, y) &= [288, 288] \end{aligned}$$

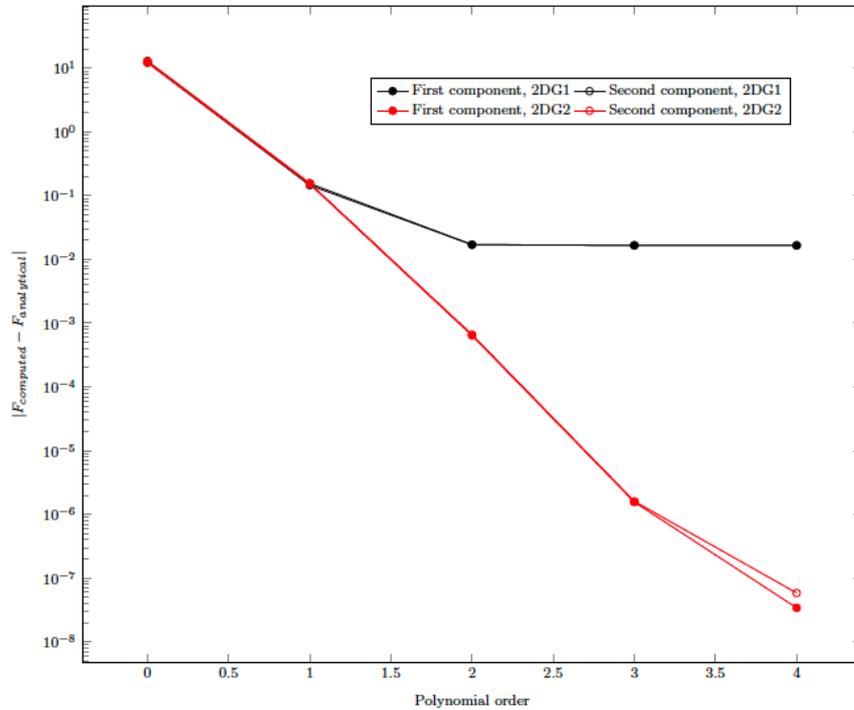


Figure 12.16 – 2D geometric convergence study.

Fig. 12.16 suggests that starting from the second polynomial order there is a significant difference in terms of accuracy if we adopt a first or a second geometric order for the computational domain. Moreover, we notice that the simulated results provided by our implementation in FEEL++ show that the accuracy for first order has a plateau without a remarkable gain for polynomial order greater than 2, which is not a desirable behavior.

2 Operator splitting convergence study

In this section we present three convergence test cases to verify the operator splitting method presented in Ch. 7. The error criteria we have adopted for all these cases are the following:

- for circuit variables

$$err_{\underline{y}} = \left(\Delta t \sum_{n=(per-1)N_{\tau}}^{per N_{\tau}} \frac{\left\| \underline{U}^{1/2} \underline{y}(t^n) - \underline{U}_{ex}^{1/2} \underline{y}_{ex}(t^n) \right\|^2}{\left\| \underline{U}_{ex}^{1/2} \underline{y}_{ex}(t^n) \right\|^2} \right)^{1/2} \quad (12.3a)$$

where \underline{y} is the vector of circuit unknowns, \underline{U} is the diagonal tensor described in Tab. 7.2, per represents the time interval of one period when the numerical solution has reached the periodic steady state, and the subscript ex denotes the exact solution;

- for 3D variables

$$err_p = \left(\Delta t \sum_{n=(per-1)N_{\tau}}^{per N_{\tau}} \frac{\|p^n - p_{ex}(t^n)\|_{L^2(\Omega)}^2}{\|p_{ex}(t^n)\|_{L^2(\Omega)}^2} \right)^{1/2} \quad (12.3b)$$

$$err_{\underline{j}} = \left(\Delta t \sum_{n=(per-1)N_{\tau}}^{per N_{\tau}} \frac{\|\underline{j}^n - \underline{j}_{ex}(t^n)\|_{L^2(\Omega)}^2}{\|\underline{j}_{ex}(t^n)\|_{L^2(\Omega)}^2} \right)^{1/2} \quad (12.3c)$$

where per represents the time interval of one period when the numerical solution has reached the periodic steady state, and the subscript ex denotes the exact solution.

Moreover, we have verified for each test case that the system has reached the periodic state with a tolerance $\varepsilon = 10^{-6}$, namely

$$\max \left\{ \frac{\|p^{[per]} - p^{[per-1]}\|_{L^2(\Omega)}^2}{\|p^{[per-1]}\|_{L^2(\Omega)}^2}, \frac{\|\underline{j}^{[per]} - \underline{j}^{[per-1]}\|_{L^2(\Omega)}^2}{\|\underline{j}^{[per-1]}\|_{L^2(\Omega)}^2}, \frac{\|\underline{U}^{1/2} \underline{y}^{[per]} - \underline{U}^{1/2} \underline{y}^{[per-1]}\|_{L^2(\Omega)}^2}{\|\underline{U}^{1/2} \underline{y}^{[per-1]}\|_{L^2(\Omega)}^2} \right\} < \varepsilon \quad (12.4)$$

In particular we verified that the time \tilde{t} , which is the minimum time that allows the unknowns to satisfy Eq. (12.4), is lower than the final computational time T :

$$\tilde{t} \ll T.$$

Remark 12. We have used the HDG method with second polynomial order for the spatial discretization since it is better suited for the proposed coupling. In this algorithm, however, other standard finite element methods can be employed for the discretization.

For all test cases the 3D domain Ω is depicted in Fig. 12.17. The number of degrees of freedom are 277230, 92410 and 116616 for the pressure p , the flux \underline{j} and the trace \hat{p} , respectively.

Remark 13. We have used a different time step for the 3D and the 0D part. We will specify in each test case the $\Delta t|_{1^{st} \text{ step}} = \Delta t$ adopted in the first step, whereas for the second step we have always imposed $\Delta t|_{2^{nd} \text{ step}} = 0.0002$.

Analytical solution and data

We describe now the data that have been used to perform the convergence test. We compare the four numerical simulations obtained using a time step

$$t_{step} = [0.02; 0.01; 0.005; 0.002]$$

with an analytical solution to the problem.

Considering the framework depicted in Fig. 12.18, to compute an analytical solution for this complex multiscale system, we start by choosing

$$p(t, \underline{x}) = \beta + \left(C_b \alpha \gamma e^{\frac{H-x_3}{kL^2}} \right) \cos(\gamma t) + \alpha \sin(\gamma t) \quad \text{in } \Omega \times [0, T] \quad (12.5a)$$

$$\underline{j}(t, \underline{x}) = \begin{bmatrix} 0 \\ 0 \\ \frac{C_b \alpha \gamma}{L^2} e^{\frac{H-x_3}{kL^2}} \cos(\gamma t) \end{bmatrix} \quad \text{in } \Omega \times [0, T] \quad (12.5b)$$

where $\underline{x} = [x_1, x_2, x_3]^T$, C_b is the value of the buffer capacitor in the circuit, H and L^2 are respectively the length in the x_3 direction and the cross-sectional area of the 3D domain, k is the permeability coefficient with $\underline{K} = k\underline{I}$ defined in Eq. (7.1), β , α and γ are constants. Using the first equation of (7.1), we compute analytically the force term

$$f(t, \underline{x}) = \alpha \gamma \cos(\gamma t) \left(1 - \frac{C_b}{kL^4} e^{\frac{H-x_3}{kL^2}} \right) - C_b \alpha \gamma^2 e^{\frac{H-x_3}{kL^2}} \sin(\gamma t).$$

Thus, we compute the values Q_I and P_I on the boundary Γ_I using (12.5b) and (12.5a), respectively:

$$\begin{aligned} Q_I(t) &= C_b \alpha \gamma \cos(\gamma t) \\ P_I(t) &= \beta + \alpha \sin(\gamma t) + C_b \alpha \gamma \cos(\gamma t) \end{aligned}$$

From Q_I and P_I we use the constitutive law of the resistor R_b to compute Π_1 :

$$\frac{P_I - \Pi_1}{R_b} = Q_I \quad (12.6a)$$

$$\Rightarrow \Pi_1 = P_I - R_b Q_I \quad (12.6b)$$

$$\Rightarrow \Pi_1(t) = \beta + \alpha \sin(\gamma t) + (1 - R_b) C_b \alpha \gamma \cos(\gamma t) \quad (12.6c)$$

We retrieve the value of the electric current passing through the capacitor C_b :

$$i_{C_b} = C_b \frac{d\Pi_1}{dt} = C_b \alpha \gamma \cos(\gamma t) - (1 - R_b) C_b^2 \alpha \gamma^2 \sin(\gamma t).$$

Imposing the Kirchhoff's law for the electric currents at node Π_1 we have

$$\begin{aligned} Q_I &= i_{C_b} + i_{R_1} \\ \Rightarrow i_{R_1} &= Q_I - i_{C_b} \\ \Rightarrow i_{R_1} &= (1 - R_b) C_b^2 \alpha \gamma^2 \sin(\gamma t). \end{aligned}$$

Then we exploit the constitutive law for R_1 to obtain

$$\Pi_2 = \Pi_1 - R_1 i_{R_1} \quad (12.7a)$$

$$\Rightarrow \Pi_2(t) = \beta + \alpha \sin(\gamma t) [1 - R_1(1 - R_b)C_b^2\gamma^2] + (1 - R_b)C_b\alpha\gamma \cos(\gamma t) \quad (12.7b)$$

with R_1 constant. Again we apply the same strategy used at node Π_1 to node Π_2 :

C_1 constitutive law:

$$\begin{aligned} i_{C_1} &= C_1 \frac{d\Pi_2}{dt} \\ \Rightarrow i_{C_1} &= C_1\alpha\gamma \cos(\gamma t) [1 - R_1(1 - R_b)C_b^2\gamma^2] + \\ &\quad + (1 - R_b)C_1C_b\alpha\gamma^2 \sin(\gamma t) \end{aligned}$$

Kirchhoff's current law:

$$\begin{aligned} i_{out} + i_{R_1} &= i_{C_1} \\ \Rightarrow i_{out} &= i_{C_1} - i_{R_1} \\ \Rightarrow i_{out} &= C_1\alpha\gamma \cos(\gamma t) [1 - R_1(1 - R_b)C_b^2\gamma^2] + \\ &\quad + (1 - R_b)C_b\alpha\gamma^2 \sin(\gamma t)(C_1 + C_b) \end{aligned}$$

Finally, imposing the constitutive law for the resistor R_{out} we obtain

$$\begin{aligned} \frac{\Pi_{out} - \Pi_2}{R_{out}} &= i_{out} \\ \Rightarrow \Pi_{out} &= \Pi_2 + R_{out} i_{out} \\ \Rightarrow \Pi_{out} &= \beta + \alpha \sin(\gamma t) [1 - R_1(1 - R_b)C_b^2\gamma^2 - R_{out}(1 - R_b)C_b\gamma^2(C_1 + C_b)] + \\ &\quad + \alpha\gamma \cos(\gamma t) [(1 - R_b)C_b + R_{out}C_1(1 - R_1(1 - R_b)C_b^2\gamma^2)]. \end{aligned}$$

We verify now that the total energy of coupled system is decaying (Eq. (7.15)) therefore granting stability of the splitting. Subsequently the discussion on the energy in Ch. 7, the only condition we have to check is that $\mathcal{U}_{ode} \geq 0 \quad \forall t \geq 0$.

Let us remind that $\mathcal{U}_{ode} = \langle \underline{y}, \underline{B} \underline{y} \rangle$ from Eq. (7.12) and $\underline{B} = \underline{U} \underline{A} - \frac{1}{2} \frac{d}{dt} \underline{U}$ from Eq. (7.11). Let now compute the matrix \underline{A} and \underline{U} for the circuit depicted in Fig. 12.18 using as vector of circuit unknowns $\underline{y} = [\Pi_1, \Pi_2]^T$:

$$\underline{A} = \begin{bmatrix} -\frac{1}{C_b R_1} & \frac{1}{C_b R_1} \\ \frac{1}{C_1 R_1} & -\frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_{out}} \right) \end{bmatrix} \quad \underline{U} = \begin{bmatrix} C_b & 0 \\ 0 & C_1 \end{bmatrix}$$

Thus, we have

$$\underline{B} = \begin{bmatrix} \frac{1}{R_1} - \frac{1}{2} \frac{dC_b}{dt} & -\frac{1}{R_1} \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_{out}} - \frac{1}{2} \frac{dC_1}{dt} \end{bmatrix},$$

which leads to

$$\mathcal{U}_{ode} = \underbrace{\frac{1}{R_1} (\Pi_1 - \Pi_2)^2 + \frac{\Pi_2^2}{R_{out}}}_I - \underbrace{\frac{\Pi_1^2}{2} \frac{dC_b}{dt}}_{II} - \underbrace{\frac{\Pi_2^2}{2} \frac{dC_1}{dt}}_{III} \quad (12.8)$$

If C_b and C_1 are constant in time, terms II and III are vanishing leaving only I , which is always positive. Thus, we have proved that $\mathcal{U}_{ode} \geq 0 \quad \forall t \geq 0$.

We eventually obtain the analytical solutions to compute the convergence order for this linear test case:

$$p = \beta + \left(C_b \alpha \gamma e^{\frac{H-x_3}{kL^2}} \right) \cos(\gamma t) + \alpha \sin(\gamma t) \quad \text{in } \Omega \times [0, T]$$

$$\underline{j} = \begin{bmatrix} 0 \\ 0 \\ \frac{C_b \alpha \gamma}{L^2} e^{\frac{H-x_3}{kL^2}} \cos(\gamma t) \end{bmatrix} \quad \text{in } \Omega \times [0, T]$$

$$f = \alpha \gamma \cos(\gamma t) \left(1 - \frac{C_b}{kL^4} e^{\frac{H-x_3}{kL^2}} \right) - C_b \alpha \gamma^2 e^{\frac{H-x_3}{kL^2}} \sin(\gamma t) \quad \text{in } \Omega \times [0, T]$$

$$Q_I = C_b \alpha \gamma \cos(\gamma t) \quad \text{for } t \in [0, T]$$

$$\Pi_1 = \beta + \alpha \sin(\gamma t) + C_b \alpha \gamma \cos(\gamma t) \quad \text{for } t \in [0, T]$$

$$\Pi_2 = \beta + \alpha \sin(\gamma t) [1 - R_1(1 - R_b)C_b^2 \gamma^2] + (1 - R_b)C_b \alpha \gamma \cos(\gamma t)$$

$$\begin{aligned} \Pi_{out} = & \beta + \alpha \sin(\gamma t) [1 - R_1(1 - R_b)C_b^2 \gamma^2 - R_{out}(1 - R_b)C_b \gamma^2 (C_1 + C_b)] + \\ & + \alpha \gamma \cos(\gamma t) [(1 - R_b)C_b + R_{out}C_1(1 - R_1(1 - R_b)C_b^2 \gamma^2)] \quad \text{for } t \in [0, T] \end{aligned}$$

In the numerical simulations presented below, we adopt the following choices for the parameter values:

$$\begin{array}{lll} \beta = 10 & \alpha = 0.5 & \gamma = 2\pi \\ k = 1 \text{ cm}^3 \text{ s Kg}^{-1} & H = 2 \text{ cm} & L = 1 \text{ cm} \\ R_{out} = 0.5 \text{ mmHg s cm}^{-3} & R_b = 2 \text{ mmHg s cm}^{-3} & R_1 = \frac{1}{4\pi^2} \text{ mmHg s cm}^{-3} \\ C_b = 1 \text{ cm}^3 \text{ mmHg}^{-1} & C_1 = 1 \text{ cm}^3 \text{ mmHg}^{-1} & T = 20 \text{ s} \end{array}$$

Results

The simulation results are illustrated in this section.

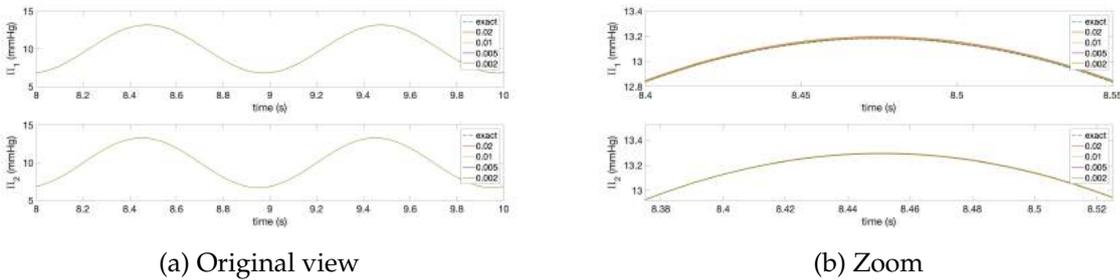


Figure 12.19 – Comparison between analytical and numerical solutions for \underline{y} .

Fig. 12.19 shows the temporal evolution of the simulation results compared to the analytical solution proposed in the previous section. To highlight the convergence results we have selected just the last periods and, moreover, for each variable we provide a zoom to point out the differences of the results obtained with various time steps. Π_2 (bottom panel) presents a better approximation from a qualitative viewpoint than Π_1 (top panel). This is expected due to

the fact that Π_1 is the key-point in the operator splitting method and it is not directly influenced by any boundary condition or external source term.

Variable	Computed order
Π_1	1.155298
Π_2	1.143255
p	1.027377
j	0.995390

Table 12.6 – Computed convergence order for the linear test case

Tab. 12.6 and Fig. 12.20 report the convergence order, which have been computed following the criteria in Eq. (12.3) and confirm the theoretical expectations (Remark 7 in Ch. 7).

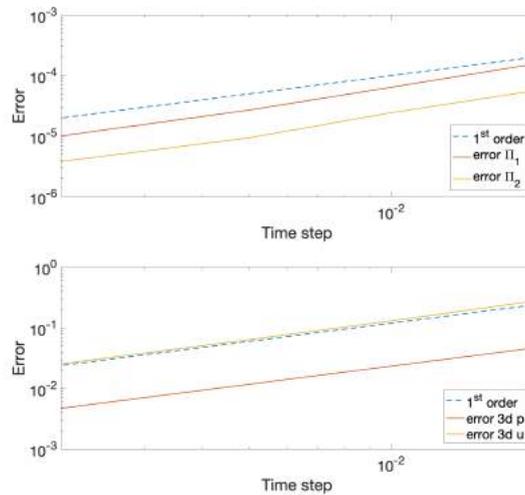


Figure 12.20 – Convergence order for the main variables for the linear test case. Top panel: circuit variables. Bottom panel: 3D variables.

2.2 Nonlinear resistor test case

In this second convergence test case for the operator splitting method we consider the same domain Ω and the same circuit Υ of the previous test with the difference that the resistor R_1 is nonlinear, in particular

$$R_1 = \bar{R}_1 + \frac{\alpha_0}{\alpha_1 + e^{-\alpha_2 \Pi_1}} \tag{12.9}$$

where \bar{R}_1 and α_0 are constant resistances, α_i , $i = 1, 2$ are positive constants and Π_1 is the unknown pressure at the associated node. The proposed nonlinearity is meaningful from a biomedical viewpoint (*e.g.* autoregulation [91]).

We impose also the same boundary conditions, and the unknowns in the lumped-parameter model are again $\underline{y} = [\Pi_1, \Pi_2]^T$ with $\underline{U} = \begin{bmatrix} C_b & 0 \\ 0 & C_1 \end{bmatrix}$.

Analytical solution and data

We compare the four numerical simulations obtained using a time step

$$t_{step} = [0.02; 0.01; 0.005; 0.002]$$

with an analytical solution to the problem.

Following the same steps as in the previous section, we can start by choosing the same $p(t, \underline{x})$, $j(t, \underline{x})$ and $\Pi_1(t)$ (Eqs. (12.5) and (12.6c)). With respect to the previous test case, the difference appears when computing Π_2 ; indeed:

$$\begin{aligned} \Pi_2 &= \Pi_1 + C_b R_1 \frac{d\Pi_1}{dt} - R_1 Q_I \\ \Rightarrow \Pi_2 &= \Pi_1 + \left(\bar{R}_1 + \frac{\alpha_0}{\alpha_1 + e^{-\alpha_2 \Pi_1}} \right) \left(C_b \frac{d\Pi_1}{dt} - Q_I \right) \end{aligned}$$

Finally imposing the Kirchhoff's law for currents on node Π_2 we obtain Π_{out} :

$$\begin{aligned} \frac{\Pi_{out} - \Pi_2}{R_{out}} &= C_1 \frac{d\Pi_2}{dt} - \frac{\Pi_1 - \Pi_2}{R_1} \\ \Rightarrow \Pi_{out} &= \Pi_2 - \frac{R_{out}}{R_1} (\Pi_1 - \Pi_2) + C_1 R_{out} \frac{d\Pi_2}{dt} \\ \Rightarrow \Pi_{out} &= \Pi_2 - \frac{R_{out}}{\bar{R}_1 + \frac{\alpha_0}{\alpha_1 + e^{-\alpha_2 \Pi_1}}} (\Pi_1 - \Pi_2) + C_1 R_{out} \frac{d\Pi_2}{dt}. \end{aligned}$$

In the context of energy estimate we can follow the same procedure as in the previous section obtaining the same expression (Eq. (12.8)). Since C_b and C_1 are constant in time, in order to have $\mathcal{U}_{ode} \geq 0 \quad \forall t \geq 0$, we have to ensure that the resistance $R_1 > 0$, *i.e.* $\alpha_0 > 0$ and $\alpha_1 > 0$.

For this test case we consider the following parameters values, which define the analytical solution:

$$\begin{array}{lll} \beta = 10 & \alpha = 0.5 & \gamma = 2\pi \\ k = 1 \text{ cm}^3 \text{ s Kg}^{-1} & H = 2 \text{ cm} & L = 1 \text{ cm} \\ R_{out} = 0.5 \text{ mmHg s cm}^{-3} & R_b = 2 \text{ mmHg s cm}^{-3} & \bar{R}_1 = \frac{1}{4\pi^2} \text{ mmHg s cm}^{-3} \\ \alpha_0 = \frac{1}{4\pi^2} \text{ mmHg s cm}^{-3} & \alpha_1 = 10 & \alpha_2 = 5 \\ C_b = 1 \text{ cm}^3 \text{ mmHg}^{-1} & C_1 = 1 \text{ cm}^3 \text{ mmHg}^{-1} & T = 10 \text{ s} \end{array}$$

Results

Fig. 12.21 shows the temporal evolution of the simulation results compared with the analytical solution proposed in the previous section. To highlight the convergence results we have selected just the last periods and, moreover, for each variable we provide a zoom to point out the differences of the results obtained with various time steps. As expected, Π_2 (bottom panel) presents a better approximation from a qualitative viewpoint than Π_1 due to the fact that Π_1 is

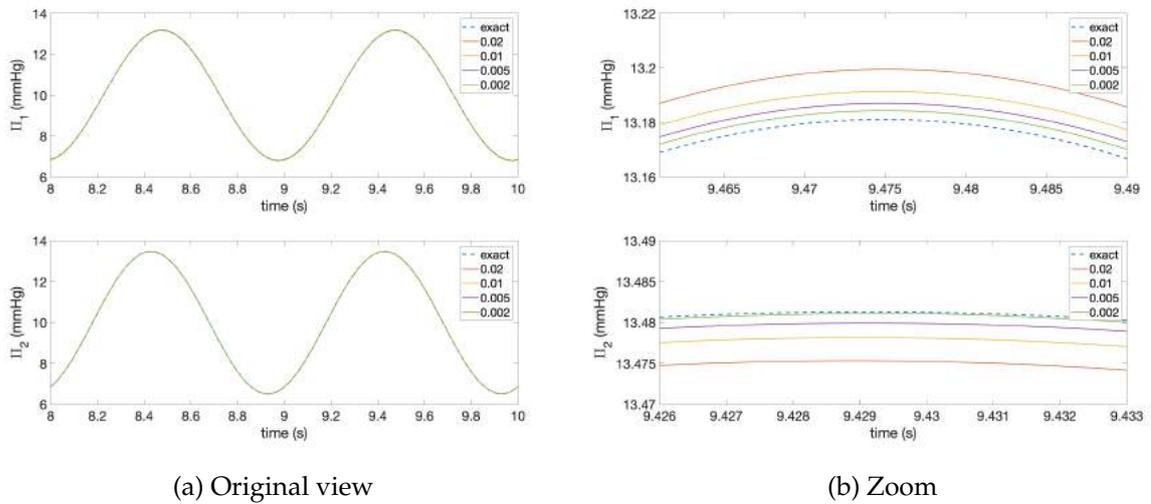


Figure 12.21 – Comparison between analytical and numerical solutions for y in the nonlinear resistor test case.

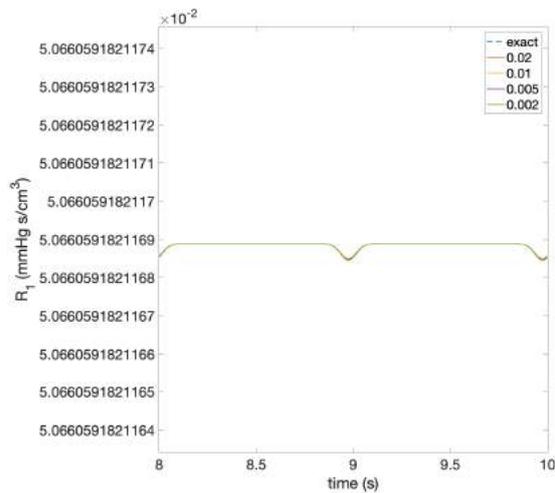


Figure 12.22 – Time profile of the nonlinear resistance value R_1 described by Eq. (12.9).

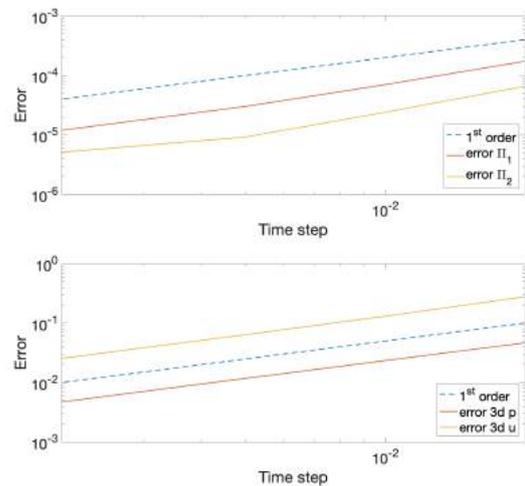


Figure 12.23 – Convergence order for the main variables for the nonlinear resistor test case. Top panel: circuit variables. Bottom panel: 3D variables.

Variable	Computed order
Π_1	1.135537
Π_2	1.132493
p	1.028557
j	0.995058

Table 12.7 – Computed convergence order for the nonlinear resistor test case.

involved directly in both steps of the operator splitting method, whereas Π_2 is an unknown just in the second step. Fig. 12.22 illustrates the computed time profile of the nonlinear resistance value R_1 , which is described by Eq. (12.9)

Tab. 12.7 and Fig. 12.23 report the convergence order, which have been computed following

the criteria in Eq. (12.3) and confirm the theoretical expectations (Remark 7 in Ch. 7).

2.3 Nonlinear resistor and capacitor test case

We extend the convergence tests to a case with the same domain and circuit of the previous study with the exception of the nonlinear capacitor

$$C_1 = \frac{\bar{C}_1}{1 + \eta \omega_2} \quad (12.10)$$

where \bar{C}_1 is a constant capacitance, η is a positive constant, and $\omega_2 = C_1 \Pi_2$ is the electric charge (analogous to fluid volume) within the capacitor C_1 . We impose the same boundary conditions and the unknowns in the circuit Υ are $\underline{y} = [\Pi_1, \omega_2]^T$ with $\underline{U} = \begin{bmatrix} C_b & 0 \\ 0 & \frac{1}{C_1} \end{bmatrix}$.

Analytical solution and data

We completed four numerical simulations using a time step

$$t_{step} = [0.005; 0.003; 0.002; 0.001].$$

Following a similar process exploited in previous test cases, we choose the same $p(t, \underline{x})$, $j(t, \underline{x})$ and $\Pi_1(t)$ (Eqs. (12.5) and (12.6c)). The difference comes up when computing Π_2 - and consequently the circuit unknown ω_2 - indeed

$$\begin{aligned} \Pi_2 &= \Pi_1 + C_b R_1 \frac{d\Pi_1}{dt} - R_1 Q_I \\ \Rightarrow \Pi_2(t) &= \Pi_1 + \left(\bar{R}_1 + \frac{\alpha_0}{\alpha_1 + e^{-\alpha_2 \Pi_1}} \right) \left(C_b \frac{d\Pi_1}{dt} - Q_I \right) \end{aligned}$$

and

$$\begin{aligned} \omega_2 &= C_1 \Pi_2 \\ \Rightarrow \omega_2 &= \frac{\bar{C}_1}{1 + \eta \omega_2} \Pi_2 \\ \Rightarrow \omega_2(t) &= \frac{-1 + \sqrt{1 + 4\eta \bar{C}_1 \Pi_2}}{2\eta} \end{aligned}$$

with $\frac{d\omega_2}{dt} = \frac{1}{2\eta} \frac{4\eta \bar{C}_1}{\sqrt{1 + 4\eta \bar{C}_1 \Pi_2}} \frac{d\Pi_2}{dt}$.

Finally imposing the Kirchhoff's law for currents on node Π_2 we obtain Π_{out} :

$$\begin{aligned} \frac{\Pi_{out} - \Pi_2}{R_{out}} &= \frac{d\omega_2}{dt} - \frac{\Pi_1 - \Pi_2}{R_1} \\ \Rightarrow \Pi_{out} &= \Pi_2 - \frac{R_{out}}{R_1(\Pi_1)} (\Pi_1 - \Pi_2) + R_{out} \frac{d\omega_2}{dt} \end{aligned}$$

We verify also in this test case that the total energy of coupled system is decaying (Eq. (7.15)) therefore granting stability of the splitting. Let us remind that $\mathcal{U}_{ode} = \langle \underline{y}, \underline{B} \underline{y} \rangle$ from Eq. (7.12)

and $\underline{\underline{B}} = \underline{\underline{U}}\underline{\underline{A}} - \frac{1}{2} \frac{d}{dt} \underline{\underline{U}}$ from Eq. (7.11). In this case the vector of circuit unknowns is $\underline{y} = [\Pi_1, \omega_2]^T$, thus we have:

$$\underline{\underline{A}} = \begin{bmatrix} 1 & 1 \\ \frac{C_b R_1}{1} & -\frac{C_1 C_b R_1}{1} \\ -\frac{1}{R_1} & \frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_{out}} \right) \end{bmatrix} \quad \underline{\underline{U}} = \begin{bmatrix} C_b & 0 \\ 0 & \frac{1}{C_1} \end{bmatrix}$$

Thus, we have

$$\underline{\underline{B}} = \begin{bmatrix} \frac{1}{R_1} - \frac{1}{2} \frac{dC_b}{dt} & -\frac{1}{C_1 R_1} \\ -\frac{1}{C_1 R_1} & \frac{1}{C_1^2} \left(\frac{1}{R_1} + \frac{1}{R_{out}} \right) - \frac{1}{2} \frac{dC_1}{dt} \end{bmatrix},$$

which leads to

$$\mathcal{U}_{ode} = \underbrace{\frac{1}{R_1} \left(\Pi_1 - \frac{\omega_2}{C_1} \right)^2}_I + \underbrace{\frac{\omega^2}{C_1^2 R_{out}} - \frac{\Pi_1^2}{2} \frac{dC_b}{dt}}_{II} - \underbrace{\frac{\omega^2}{2} \frac{dC_1}{dt}}_{III} \quad (12.11)$$

As the nonlinear resistor test case we impose that $R_1 > 0$, *i.e.* $\alpha_0 > 0$ and $\alpha_1 > 0$, in order to ensure that contribution *I* is positive. Contribution *II* is vanishing since C_b is constant. Considering Eq. (12.10), we can compute $\frac{dC_1}{dt} = -\frac{\eta}{(1 + \eta\omega_2)^2} \frac{d\omega_2}{dt}$. Thus, if $\eta > 0$ and $\frac{d\omega_2}{dt} \geq 0$, contribution *III* is positive.

Under these assumptions, we have proved that $\mathcal{U}_{ode} \geq 0 \quad \forall t \geq 0$.

For this test case we consider the following parameters value set, which define the analytical solution:

$$\begin{array}{lll} \beta = 10 & \alpha = 0.5 & \gamma = 2\pi \\ k = 1 \text{ cm}^3 \text{ s Kg}^{-1} & H = 2 \text{ cm} & L = 1 \text{ cm} \\ R_{out} = 0.5 \text{ mmHg s cm}^{-3} & R_b = 2 \text{ mmHg s cm}^{-3} & \bar{R}_1 = \frac{1}{4\pi^2} \text{ mmHg s cm}^{-3} \\ \alpha_0 = \frac{1}{4\pi^2} \text{ mmHg s cm}^{-3} & \alpha_1 = 10 & \alpha_2 = 5 \\ T = 10 \text{ s} & C_b = 1 \text{ cm}^3 \text{ mmHg}^{-1} & \\ \bar{C}_1 = 1 \text{ cm}^3 \text{ mmHg}^{-1} & \eta = 5 & \end{array}$$

Results

The nonlinear resistor is the same as the previous test case (Fig. 12.22), whereas the nonlinear capacitance time profile is shown in Fig. 12.25.

Fig. 12.24 shows the temporal evolution of the simulation results compared with the analytical solution proposed above. To emphasize the convergence results we have selected just the last periods and, moreover, for each variable we provide a zoom to point out the differences of the results obtained with various time steps. As expected, ω_2 (bottom panel) presents a better approximation from a qualitative viewpoint than Π_1 due to the fact that Π_1 is involved directly in both steps of the operator splitting method, whereas ω_2 is an unknown just in the second

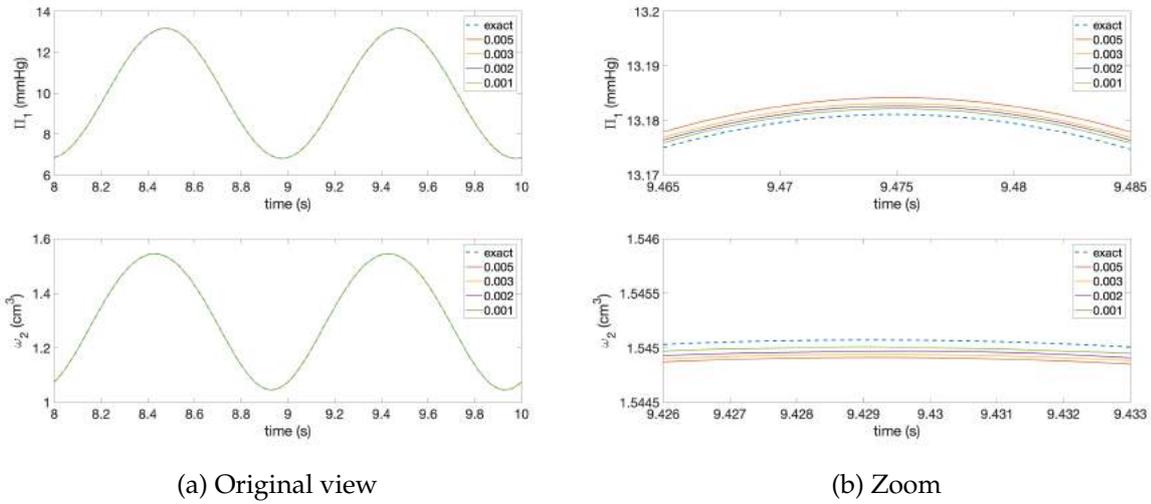


Figure 12.24 – Comparison between analytical and numerical solutions for \underline{y} in the nonlinear resistor and capacitor test case.

step. Note that in this test case $\frac{d\omega_2}{dt} \geq 0$, however the method still converges. The theorem indeed calls for a condition - the positivity of $\frac{d\omega_2}{dt}$ - that is sufficient, but not necessary for the convergence of the method.

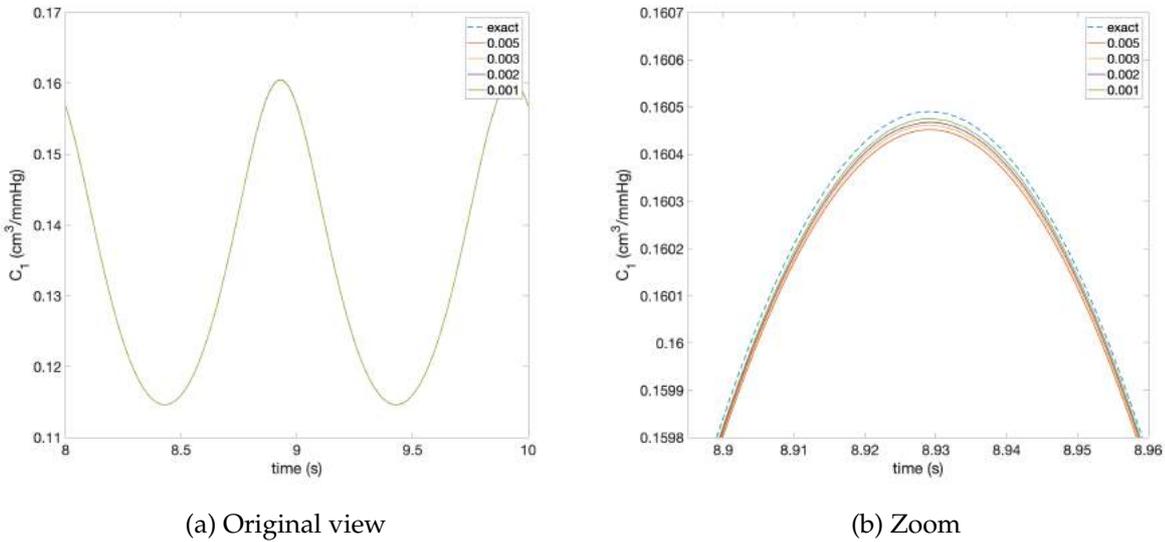


Figure 12.25 – Time profile of the nonlinear capacitance value C_1 described by Eq. 12.10.

Variable	Computed order
Π_1	0.925018
ω_2	0.928083
p	1.002411
\underline{j}	0.998717

Table 12.8 – Computed convergence order for the nonlinear resistor and capacitor test case.

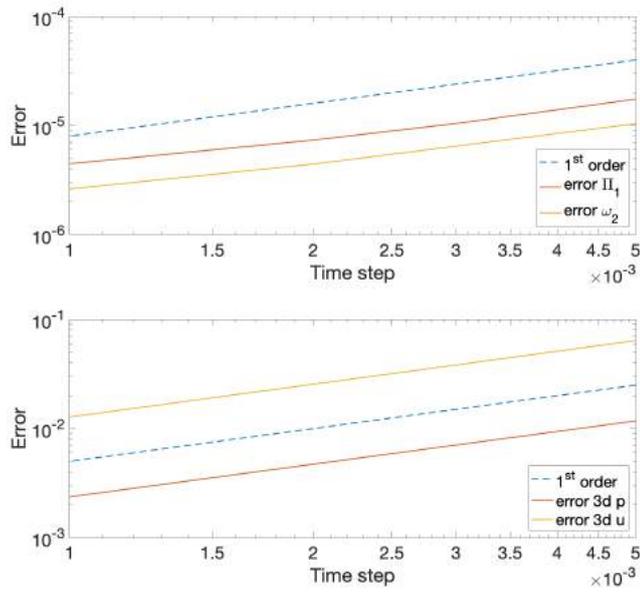


Figure 12.26 – Convergence order for the main variables for the nonlinear resistor and capacitor test case. Top panel: circuit variables. Bottom panel: 3D variables.

Tab. 12.8 and Fig. 12.26 report the convergence order, which have been computed following the criteria in Eq. (12.3) and confirm the theoretical expectations (Remark 7 in Ch. 7).

Contents

1	Hemodynamics of the ocular posterior segment	160
2	Biomechanics of the lamina cribrosa	162
3	Biomechanics of cornea, sclera, choroid and retina.	164

In this chapter we discuss the validation of the OMVS. We compare our simulation results with clinical or experimental data whenever available, and also review similar outcomes in the literature.

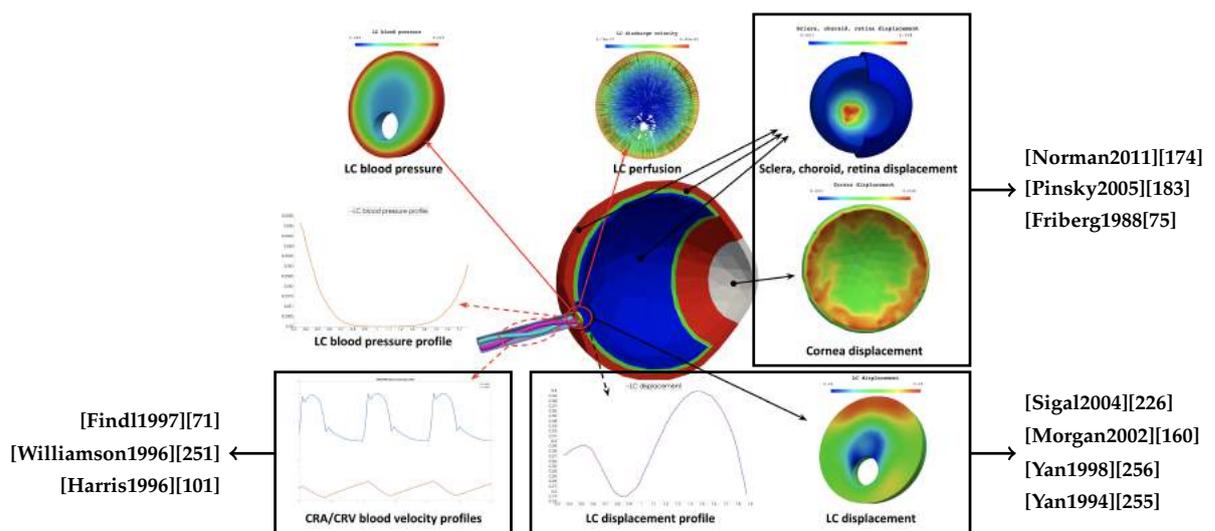


Figure 13.1 – Overview of the OMVS validation.

Fig. 13.1 recaps all the output given by *Level 2b* of the OMVS (Ch. 5) and lists the main

sources where we retrieved the data for the validation.

We highlight the fact that for the hemodynamics within the lamina cribrosa to the best of our knowledge there are no experimental or clinical data, thus we cannot directly compare the simulation results; however, since the system is all coupled we consider that if all the other parts reproduce a physiological behavior, also the lamina cribrosa perfusion computed by the OMVS should attain realistic values.

We provide in the next sections more details on the data comparison.

1 Hemodynamics of the ocular posterior segment

The validation of the central retinal vessels hemodynamics, *i.e.* the circuit presented in Ch. 3, is based on the work published by Guidoboni et al. in 2014 on IOVS [91]. The paper shows that, despite the numerous simplifying assumption in the simple 0D scheme, the proposed mathematical model is able to capture the mechanical action of IOP on clinically measurable hemodynamic quantities [101, 71, 251], in particular the total retinal blood flow and the CRA blood velocity. The results obtained in that study were used to assess the independent effects of varying arterial blood pressure and blood flow autoregulation on the relationship between IOP and blood flow. Notably, the model predicts that the steep decay in retinal blood flow that happens due to high IOP would shift toward higher values as the blood pressure of the subject increases. This outcome agrees with the clinical study of He et al. [104], who found that a higher IOP was needed to attenuate ocular blood flow in Long-Evans rats with higher systemic blood pressure.

These theoretical predictions on the relationship between intraocular pressure, blood pressure, ocular perfusion and glaucoma have been confirmed by the Singapore Epidemiology of Eye Diseases study, an independent population-based study including nearly 10000 individuals [237].

In another validation test, we utilize the values of IOP and systemic systolic and diastolic blood pressure (SP/DP) reported in [51] as input data for our model and we compare the predictions of the simulation with the clinical results provided in the paper. For this comparative study we have employed *Level 2a* of the OMVS (Ch. 5). We have simulated three test case, which data are resumed in Tab. 13.1:

- (i) systemic hypertensive patients (HBP) with high IOP;
- (ii) system hypertensive patients (HBP) with normal IOP;
- (iii) normal systemic blood pressure patients (NBP) with high IOP.

TEST CASE	SP[mmHg]	DP[mmHg]	HR[beats/min]	IOP[mmHg]
<i>HBP - High IOP</i>	170	100	60	28
<i>HBP - Normal IOP</i>	170	100	60	15
<i>NBP - High IOP</i>	139.2	82.4	60	28

Table 13.1 – Input data for the comparative study with [51].

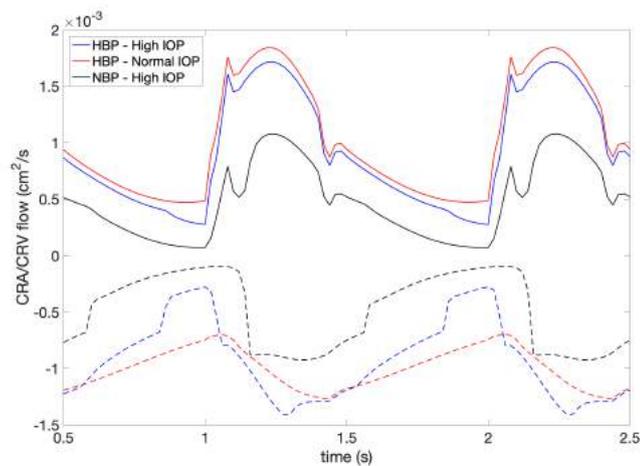


Figure 13.2 – CRA/CRV blood flow computed by the OMVS using the data retrieved from [51]

Fig. 13.2 shows the blood flow computed by the OMVS within the CRA and CRV. Systemic hypertensive virtual patients share a similar blood flow waveform within the CRA (red and blue solid lines), whereas our model suggests a lower CRA blood flow for subject *NBP - High IOP* (black solid line). For what concerns the CRV, the simulation results for virtual patients with elevated IOP (blue and black dashed lines) show a drop in the blood flow with respect to *HBP - Normal IOP* (red dashed line). This reduction lasts longer in the case of *NBP - High IOP* (black dashed line).

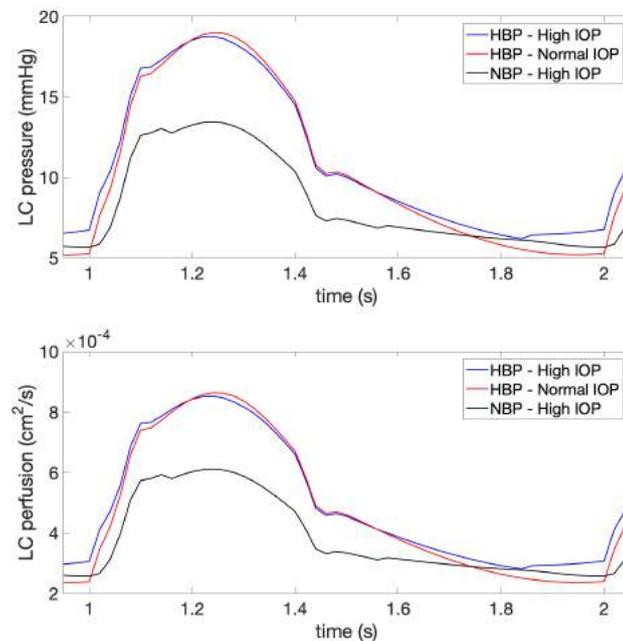


Figure 13.3 – Lamina cribrosa hemodynamics computed by the OMVS using the data retrieved from [51].

Fig. 13.3 suggests that virtual patient *HBP - Normal IOP* (red line) experiences a similar hemodynamic behavior as *HBP - High IOP* (blue line), whereas the virtual patient *NBP - High IOP* (black line) exhibits a drop in the lamina cribrosa perfusion (up to 28%) .

The clinical results reported by Costa et al. [51] suggest that only individuals with high IOP and normal blood pressure may be at higher risk for glaucomatous damage. The OMVS conjectures that elevated IOP has a significant impact on the lamina cribrosa hemodynamics when combined with normal systemic blood pressure, whereas it is remarkably less noticeable in case of high systemic blood pressure. If we consider a deficit in the lamina cribrosa perfusion as a risk factor for glaucoma, which is an increasingly supported idea in ophthalmology [125, 41], the simulation results obtained employing our mathematical model are consistent with the clinical analysis presented above.

In a different study, Samsudin et al. [214] analyzed the hemodynamics in normal tension glaucoma (NTG) patients, concluding that there is no difference in the blood flow within the ophthalmic arteries between patients with NTG and healthy subjects, suggesting that vascular insufficiency or dysregulation by themselves may not account for the pathogenesis of NTG. These outcomes seem to be in contrast with the work of Costa et al., however we remind that glaucoma is a multifactorial disease and, despite the great effort in ophthalmology, there are still many mechanisms that need to be unveiled especially on the vascular system and its interaction with the biomechanics of the surrounding tissues. Thus, some biomechanical parameters - that are not measured or defined in the aforementioned papers - may be responsible for significant contributions that may lead to the development of glaucoma. In this context, our mathematical model may provide a different perspective and disentangle the effect of single risk factors in a virtual environment.

TEST CASE	SP[mmHg]	DP[mmHg]	HR[beats/min]	IOP[mmHg]
<i>NTG patient</i>	149.9	85.9	60	11.2
<i>Healthy</i>	152.1	90.6	60	11.1

Table 13.2 – Input data for the comparative study with [214].

Tab. 13.2 summarizes the input data directly retrieved from [214]. Also for this comparative study we have employed *Level 2a* of the OMVS (Ch. 5).

Fig. 13.4a shows the blood flow computed by the OMVS in the ophthalmic artery (OA) and suggests that there are no significant differences between the two simulations (1.5%). This small discrepancy is found also for the CRA/CRV blood flow (Fig. 13.4b) and for the lamina cribrosa perfusion (Fig. 13.4c), in particular it is 1.4%, 1.21% and 1.26% for CRA, CRV and lamina cribrosa, respectively. The simulation results provided by our model are in agreement with the analysis completed by Samsudin et al., indeed the hemodynamics in the *NTG patient* is very similar to the *Healthy* one.

2 Biomechanics of the lamina cribrosa

We analyze now the biomechanics of the lamina cribrosa. The validation of this part of the model is based on the works by Causin et al. [34] and Guidoboni et al. [90]. In [90] the authors compare the simulation results with experimental measurements [160, 71] suggesting that the biomechanics in the lamina has a significant impact also on the hemodynamics in the central retinal vessels. In Ch. 15 we detail a virtual experiment that investigates the effect of IOP on the biomechanics and hemodynamics within the lamina cribrosa. The results obtained employing the OMVS are consistent with the findings reported in [90]. Causin

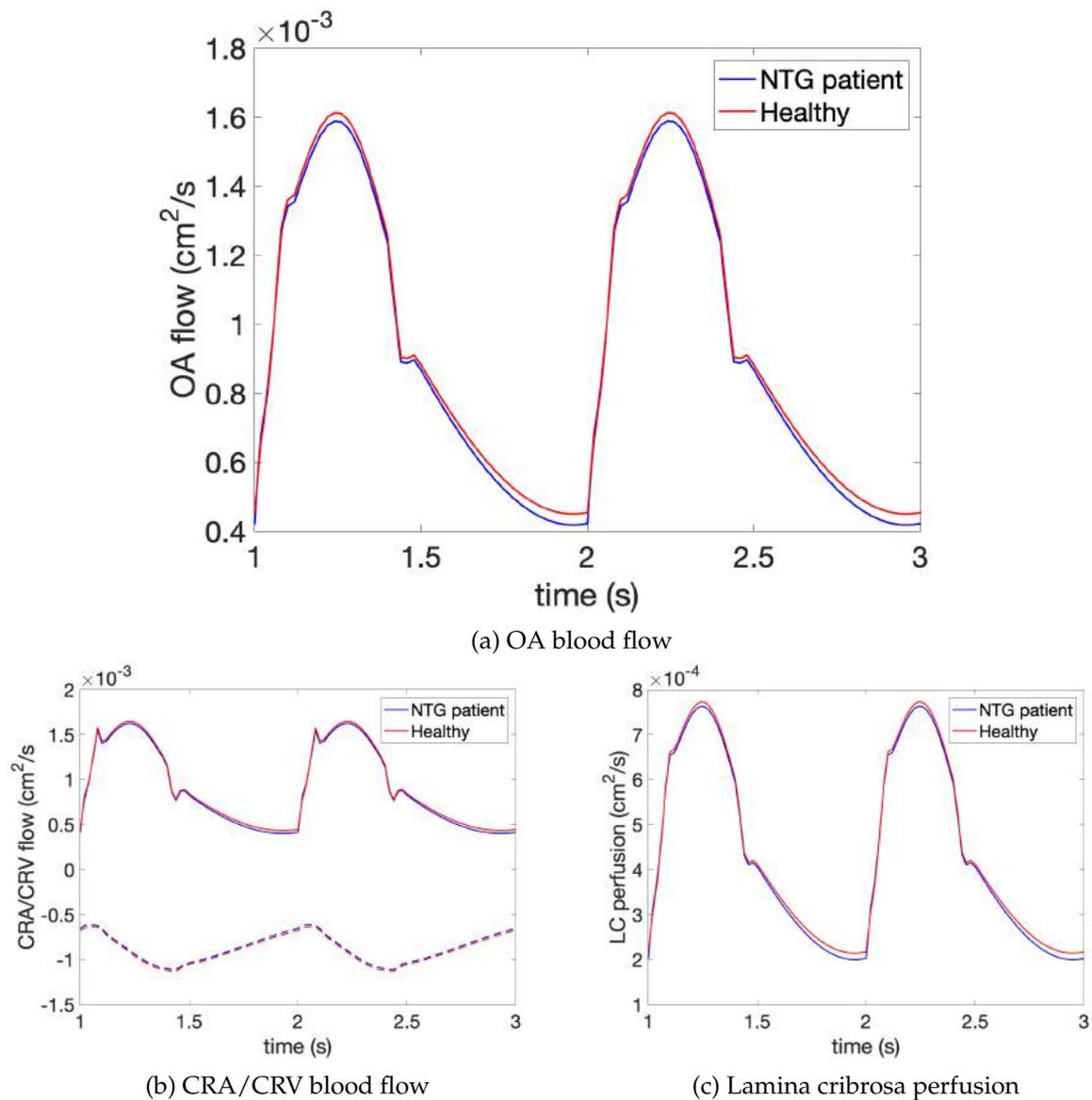


Figure 13.4 – Simulation results obtained using the data retrieved from [214].

et al., on contrary, compared their simulations - two different numerical case studies - with the experiments conducted by Yan and collaborators [256], who mounted three enucleated human eyes on a specially designed experimental apparatus, which allowed to sequentially increase the IOP. We replicated this virtual experiment using *Level 2a* of the OMVS, completing 5 test with increasing IOP (5, 15, 25, 35 and 50 mmHg). Following the data reported in [34] and [256], for all virtual tests we imposed a systemic SP/DP blood pressure of 126.3/84.2 mmHg, heart rate of 60 beats per minute, and a $RLTp = 7$ mmHg. For the elastic properties we have used the baseline values described in Sec. 3, Ch. 5.

In Fig. 13.6a we show the output 3D geometry of the lamina cribrosa using 5 different input values for the IOP, in particular we have selected the one with $IOP = 5$ mmHg as the reference value (in gray). For completeness, we illustrate also an example of output for the LC displacement for the simulation with $IOP = 15$ mmHg (Figs. 13.5a and 13.5b).

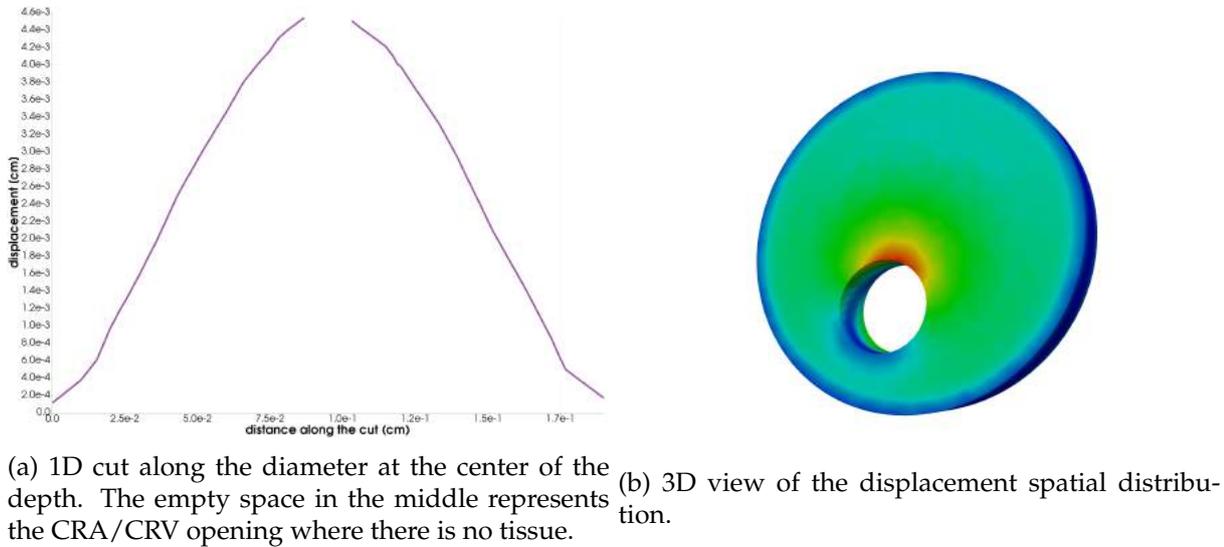


Figure 13.5 – Lamina cribrosa displacement field using baseline values.

LC displacement values (Fig. 13.5a) are comparable with the experimental studies reported in [255]. Fig. 13.5b shows that the LC biomechanics is not highly influenced by the presence of the CRA/CRV opening as pointed out in [226]. Therefore, we computed the LC volume below reference, namely ΔV , and we compared our 5 simulations with the measurements using three enucleated human eyes [256] and the results provided by [34] in the two numerical test cases. Fig. 13.6b highlights the fact that the OMVS achieves physiological biomechanical results within the lamina cribrosa when comparing the response to different levels of IOP.

3 Biomechanics of cornea, sclera, choroid and retina.

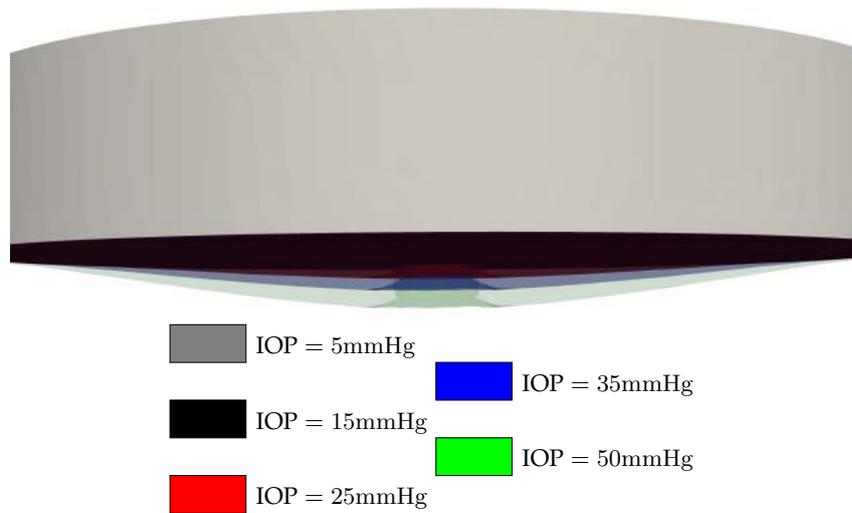
In this section we discuss the biomechanical simulation provided by the OMVS and their comparison with real data.

As a premise, we mention the fact that few works have been done in this sense, indeed the possibility to measure the stresses and the displacements experienced by retina, choroid, sclera and cornea *in vivo* is very challenging; on the other hand, many studies tried to replicate the behavior of such tissues with phantoms or *ex vivo* experiments. For this reason, we have compared the simulations results of the sclera and the cornea, which are the external and more accessible tissues, whereas for the biomechanics of the retina and choroid we plan to conclude this exhaustive analysis when more data will be available in the future.

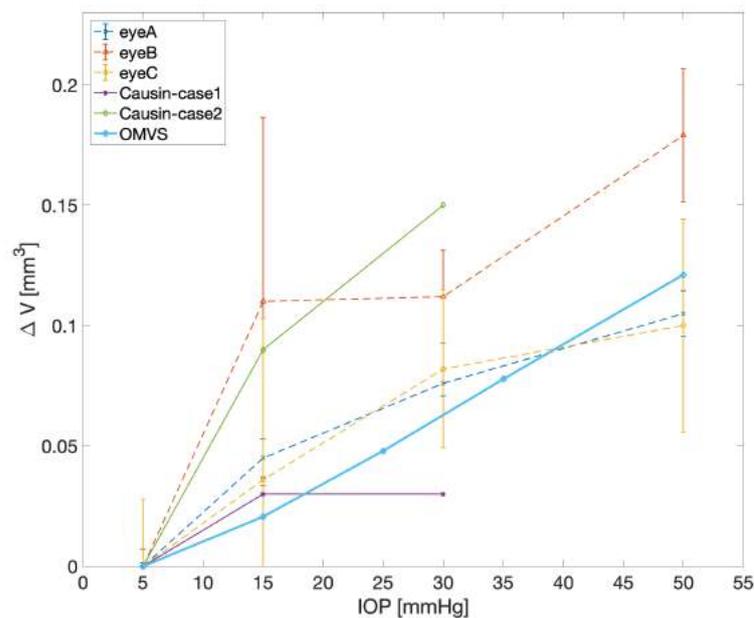
First, we refer to the work by Myers et al. [164], where the authors are performing an inflation test on a posterior bovine sclera. This *in vitro* analysis exhibits a nonlinear response to controlled pressurization. We completed 4 virtual experiments with increasing IOP (10.5, 14.78, 23.3 and 30 mmHg) employing *Level 2b* of the OMVS (Ch. 5). For all test cases we have used the biomechanical parameters presented in Tab. 5.3 and the following input data

$$SP = 126.3 \text{ mmHg} \quad DP = 84.2 \text{ mmHg} \quad HR = 60 \text{ beats/min} \quad RLTp = 7 \text{ mmHg}.$$

Myers et al. affirm that the sclera can be modeled as a quasilinear elastic material in the physiological pressure range of 2 – 6kPa, behavior which is well replicated by our virtual experiments



(a) 3D comparison among the original relaxed LC and the three OMVS simulations obtained imposing different IOP.



(b) Comparison among the IOP-induced increments of LC volume below reference (ΔV) measured experimentally by [256], numerically by [34], and our simulation results.

Figure 13.6 – Comparison of experimental and simulation results within the lamina cribrosa.

resumed in Fig. 13.7.

Second, we refer to the correlation between axial length and intraocular pressure discussed in the work by Detorakis and Pallikaris [61]. In this case we have performed 6 virtual experiments with increasing IOP (1.5, 3.83, 10.5, 14.78, 23.3 and 30 mmHg) employing *Level 2b* of the

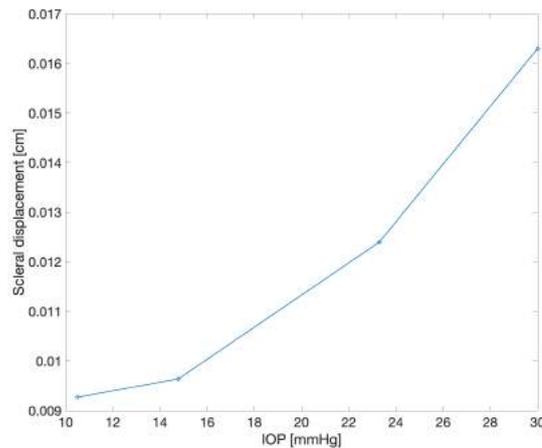


Figure 13.7 – Scleral biomechanical response at increasing IOP simulated *via* the OMVS.

OMVS (Ch. 5). For all test cases we have used the biomechanical parameters presented in Tab. 5.3 and the following input data

$$SP = 126.3 \text{ mmHg} \quad DP = 84.2 \text{ mmHg} \quad HR = 60 \text{ beats/min} \quad RLTp = 7 \text{ mmHg}.$$

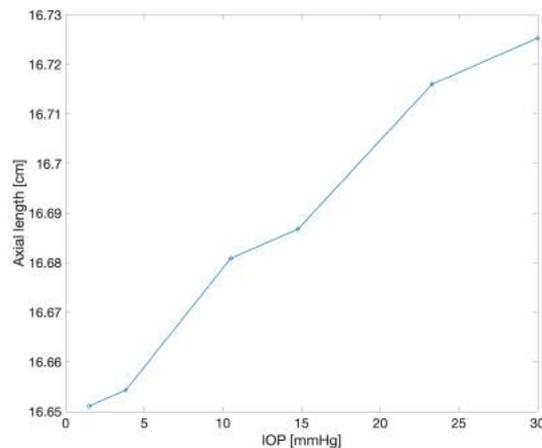


Figure 13.8 – Axial length response at increasing IOP simulated *via* the OMVS.

The authors call for a linear dependency between the measurements of intraocular pressure and axial length. The results provided by the OMVS (Fig. 13.8) are in agreement with such a correlation between these two quantities.

Finally, we compare the results computed by the OMVS with the findings of Boyce et al. [20] on the corneal displacement. In their study, the authors have conducted experiments on a bovine cornea to capture the displacement under constrained inflation conditions. We conducted 4 virtual experiments with increasing IOP (10.5, 14.78, 23.3 and 30 mmHg) employing *Level 2b* of the OMVS (Ch. 5). For all test cases we have used the biomechanical parameters presented in Tab. 5.3 and the following input data

$$SP = 126.3 \text{ mmHg} \quad DP = 84.2 \text{ mmHg} \quad HR = 60 \text{ beats/min} \quad RLTp = 7 \text{ mmHg}.$$

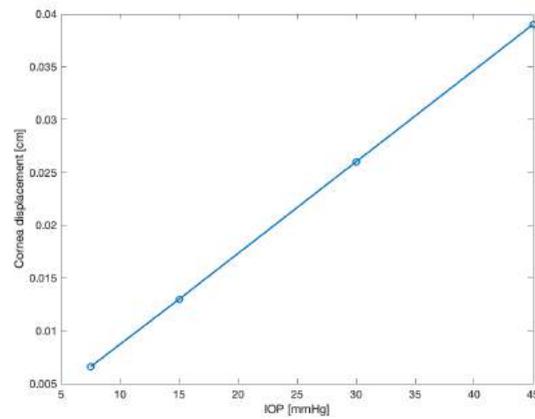


Figure 13.9 – Corneal biomechanical response at increasing IOP simulated *via* the OMVS.

Boyce et al. found that, if the IOP applied on the cornea is in physiological ranges, the displacement has a linear behavior (Figs. 7,9 and 10 of [20]). Our model predicts a similar behavior for the human cornea as illustrated by Fig. 13.9. This simulation's trend is somehow unforeseen due to the presence of several nonlinearities in the biomechanical interaction between the ocular tissues and in the hemodynamics of the ocular posterior segment.

Blood pressure study

Contents

1	Data	169
2	Results	170
3	Conclusions	172

In this chapter we employ the OMVS as a virtual laboratory to study the lamina cribrosa perfusion parameters, which are difficult to estimate with non-invasive measurements and are affected by many factors that vary among individuals and cannot be easily isolated. The content of this chapter has been published in [211]. In particular, we compare 3 virtual patients that have different blood pressure.

1 Data

For this virtual experiment we have employed the *Level 2a* of the OMVS described in Ch. 5. In particular, we simulate

- (i) the blood circulation in the retinal vasculature and the central retinal vessels via a circuit-based model;
- (ii) the blood perfusion within the lamina cribrosa via a porous-media model;
- (iii) the biomechanical behavior of the lamina cribrosa via an isotropic linear elasticity system.

We select three virtual patients, which input parameters are resumed in Tab. 14.1. Let LBP be the patient with low blood pressure, NBP the one who serves as baseline, and HBP the

subject who suffers from hypertension.

Parameter	LBP	NBP	HBP	Unit
SP	100	120	140	mmHg
DP	70	80	90	mmHg
IOP	15	15	15	mmHg
RLTp	7	7	7	mmHg

Table 14.1 – Synthetic data of the patients who have been used in the BP virtual study.

2 Results

We present the results we have obtained by employing the OMVS in this virtual test case:

- CRA and CRV blood velocities (Fig. 14.1a and 14.1b, respectively) are comparable to those obtained via direct imaging modalities [251].

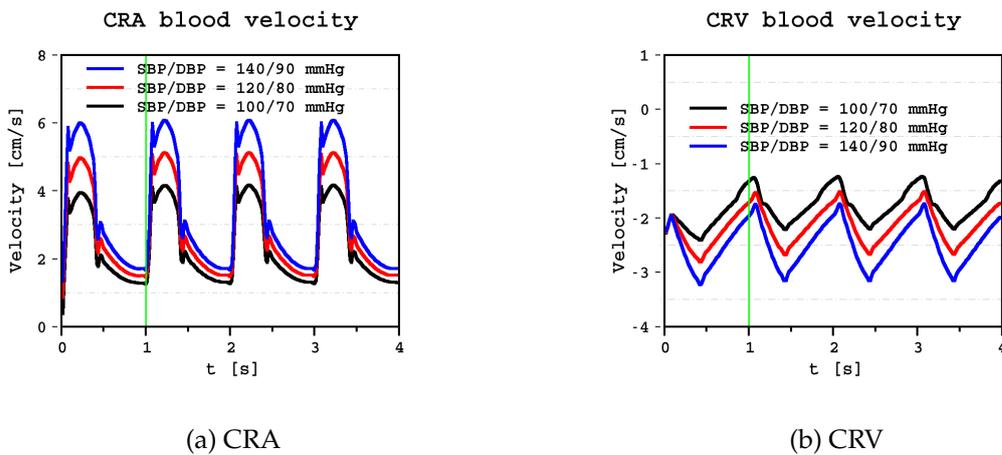


Figure 14.1 – Blood velocities time profile comparison between the 3 different virtual subjects within the central retinal vessels. The green line points out the instants $t = 1.01$ s.

- LC blood pressure distribution (Fig. 14.2) attains realistic values [91], in particular the computed arterial pressure on the lateral boundary; moreover, the simulation results suggest also that this blood pressure is highly influenced by the values of SP/DP imposed at the inlet of the circuit model.
- LC blood perfusion increases with SP/DP (Fig. 14.3), particularly near the nasal area.
- LC displacement values (Fig. 14.4) are comparable to those reported in experimental studies [255] and the simulation results suggest that the displacement field is not highly influenced by the presence of the CRA/CRV opening as reported in other studies [226].

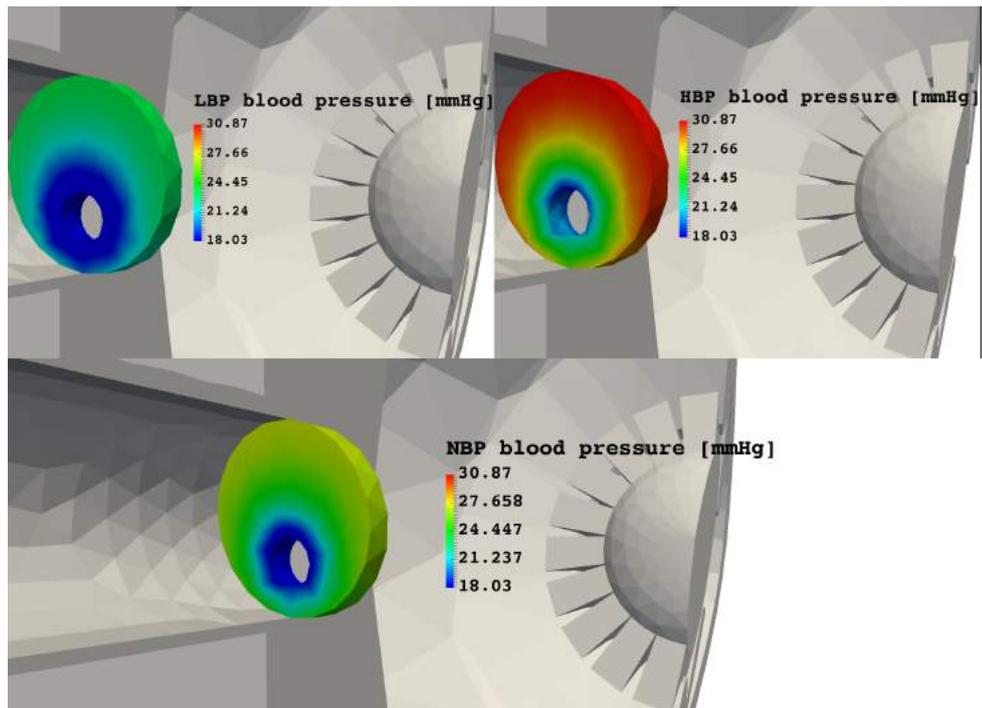


Figure 14.2 – Comparison of the lamina cribrosa blood pressure between the 3 different virtual subjects at time $t = 1.01$ s.

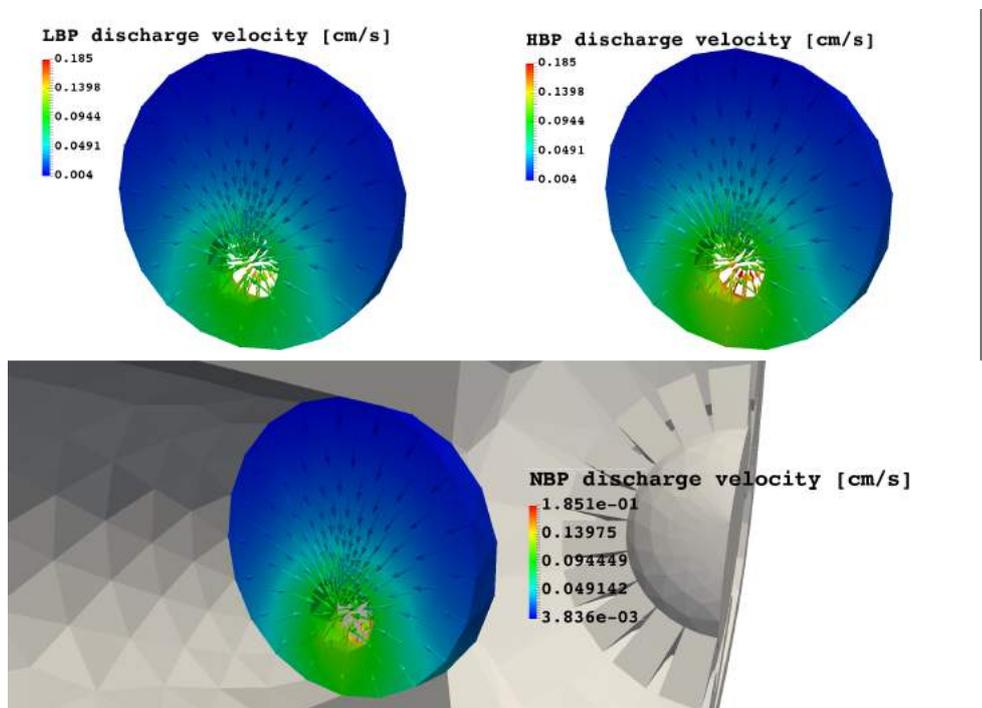


Figure 14.3 – Comparison of the lamina cribrosa blood perfusion between the 3 different virtual subjects at time $t = 1.01$ s.

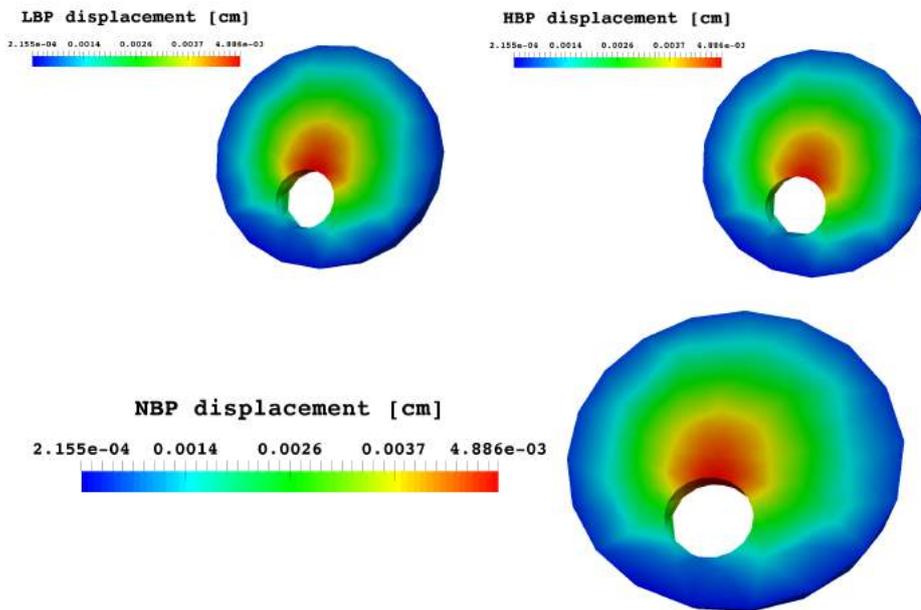


Figure 14.4 – Comparison of the lamina cribrosa displacements between the 3 different virtual subjects at time $t = 1.01$ s.

3 Conclusions

We have shown how the OMVS may serve as an instrument to illustrate and estimate lamina cribrosa perfusion parameters and predict their spatial variability and temporal evolution. Moreover, this mathematical model may provide new means to address the increasing demand of information on parts of the eye that are not easily accessible with standard investigation instruments.

Intraocular pressure study

Contents

1	Data	173
2	Results	174
3	Conclusions	175

In this chapter we present a virtual experiment that has been realized using synthetic data based on clinical measurements. Its contents has been also published in [210]. This virtual experiment compares two subjects who share the same input parameters with the exception of the IOP in order to understand the effect of this difference on the ocular hemodynamics and biomechanics.

The underlying intention is to show the predictive power of mathematical modeling in order to have quantitative information in parts of the eye that are not accessible with standard investigation tools and try to unveil the physiological mechanisms behind major diseases, such as glaucoma.

1 Data

We select two virtual patients, namely Patient 1 and Patient 2, which input parameters are resumed in Tab. 15.1. We highlight the values of IOP that are 16 mmHg for Patient 1 and 25 mmHg for Patient 2.

For this virtual experiment we have employed the *Level 2b* of the OMVS described in Ch. 5. In particular, we simulate

Parameter	Patient 1	Patient 2	Unit	Reference
IOP	16	25	mmHg	[50]
RLTp	7	7	mmHg	[203]
SP	120	120	mmHg	[254]
DP	80	80	mmHg	[254]
HR	60	60	bpm	[23]

Table 15.1 – Synthetic data of the patients that have been used in the IOP virtual study.

- (i) the blood circulation in the retinal vasculature and the central retinal vessels via a circuit-based model;
- (ii) the blood perfusion within the lamina cribrosa via a porous-media model;
- (iii) the biomechanical behavior of cornea, sclera, choroid, retina and lamina cribrosa via an isotropic linear elasticity system.

We inserted these values in our model using the platform described in Ch. 11 as illustrated by Fig. 15.1.

Figure 15.1 – Input data interface for the two virtual patients

2 Results

We present the results we have obtained by employing the OMVS in this virtual test case. For what concerns the central retinal vessels, the blood velocity profiles are highly influenced by the IOP, especially the CRV collapses if the IOP is higher than the blood pressure within the vein (Figs. 15.2a and 15.2b) as described by Eq. (3.8).

In the lamina cribrosa we point out a difference up to 2% in the blood pressure distribution and up to 5.5% in the tissue perfusion between the two patients (Figs. 15.3a and 15.3).

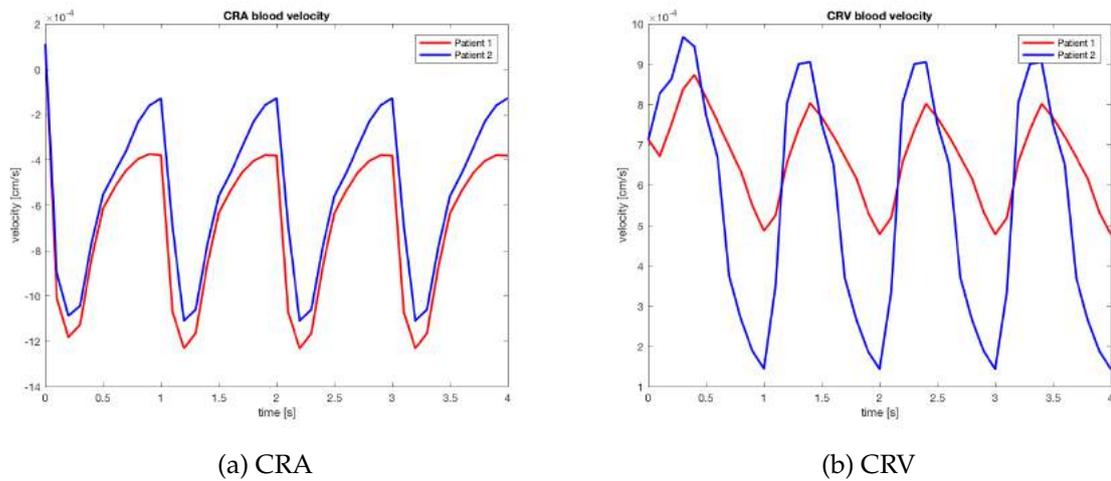
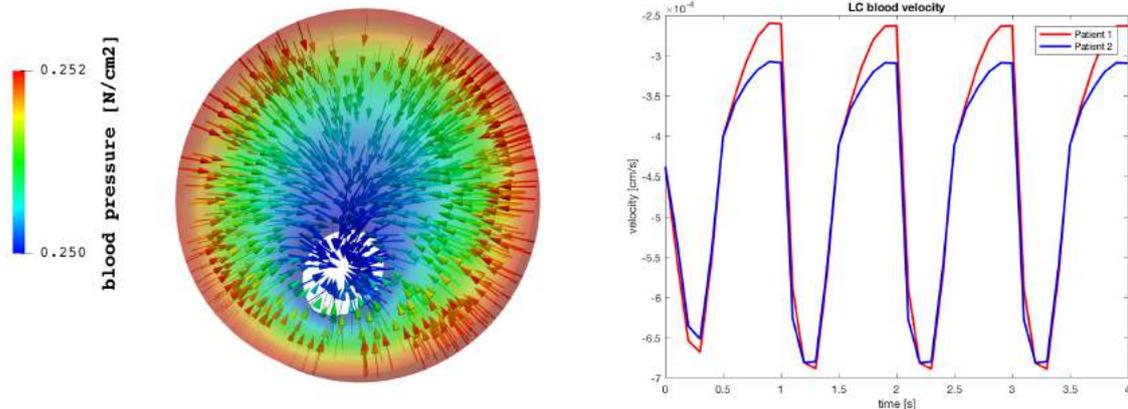


Figure 15.2 – Blood velocity profiles simulated *via* the OMVS.



(a) Blood pressure spatial distribution and blood flow orientation

Figure 15.3 – Blood velocity profile

Figure 15.4 – Hemodynamics results within the lamina cribrosa simulated *via* the OMVS

For a biomechanics viewpoint, our model predicts a difference up to 33% in the lamina cribrosa displacement distribution between the two patients with an applied strain that is doubled in Patient 2 (Figs. 15.5a and 15.5b);

Finally, we analyze the biomechanics of the other simulated tissues. We don't point out any remarkable difference between the two patients for the displacement in the sclera, choroid and retina (Fig. 15.6), whereas Fig. 15.7 shows a higher displacement along the cornea border for Patient 2.

3 Conclusions

In conclusion for the analyzed virtual study, the OMVS suggests that:

- high IOP causes CRV to collapse with a consequent influence on the overall system;

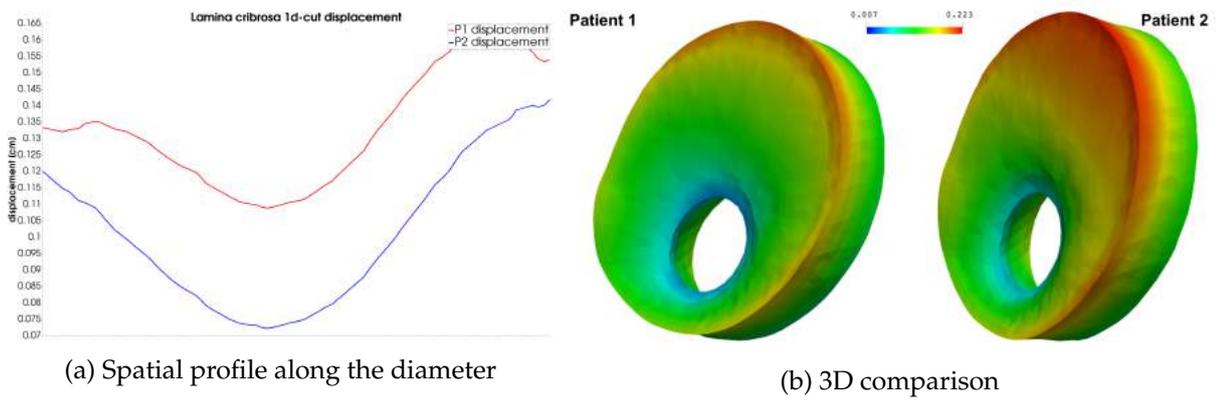


Figure 15.5 – Lamina cribrosa displacement simulated *via* the OMVS.

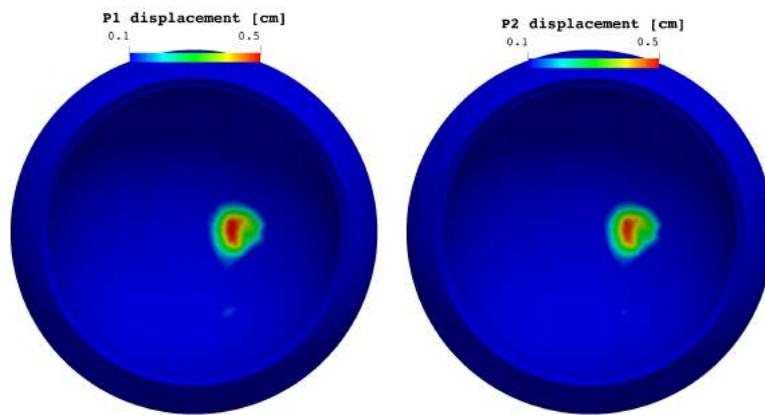


Figure 15.6 – Displacement distribution in the choroid and the retina simulated *via* OMVS.

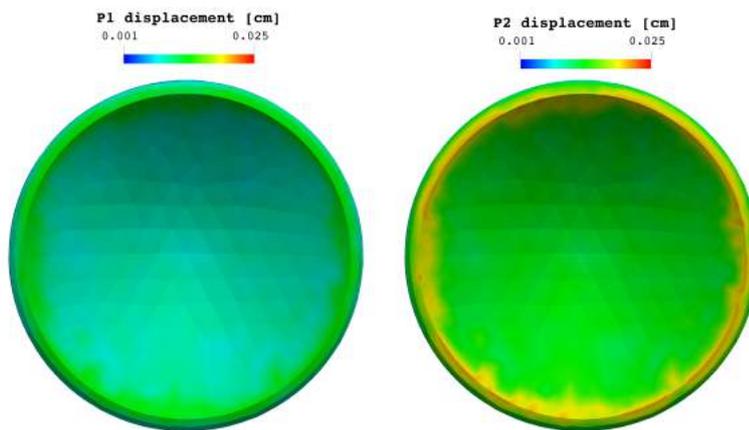


Figure 15.7 – Displacement distribution in the cornea simulated *via* OMVS.

- differences in IOP may lead also to significant differences in the lamina cribrosa blood perfusion (5.5%);
- differences in IOP lead, as expected, to significant differences in the displacement (up to 33% in the lamina).

Trans-laminar pressure difference study

Contents

1	Data	177
2	Results	178
3	Conclusions	180

In this chapter we present a virtual experiment that has been realized using synthetic data based on clinical measurements. The content of this virtual experiment has been published in [212]. We compare 5 different virtual patients in order to understand the effect of varying the translaminar tissue pressure difference on the hemodynamics and biomechanics within the lamina cribrosa.

We remind from Ch. 2 that the trans-laminar tissue pressure difference is the difference between the intraocular pressure and the retrolaminar tissue pressure (Fig. 16.1):

$$TLPd = IOP - RLTp$$

1 Data

For this virtual experiment we have employed the *Level 2b* of the OMVS described in Ch. 5. In particular, we simulate

- (i) the blood circulation in the retinal vasculature and the central retinal vessels *via* a circuit-based model;
- (ii) the blood perfusion within the lamina cribrosa *via* a porous-media model;

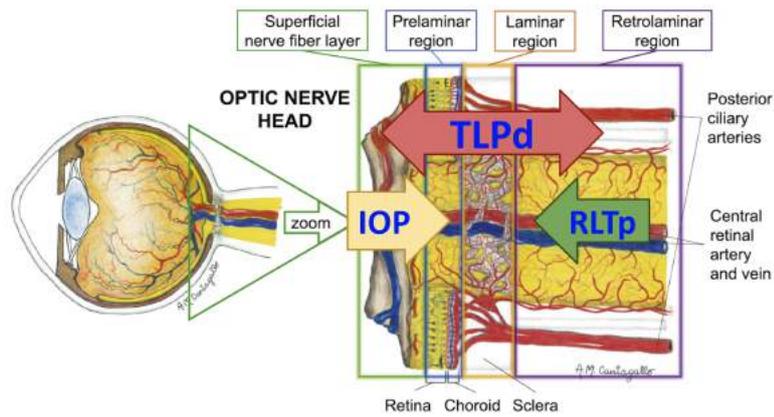


Figure 16.1 – Relationship between IOP, RLTp and TLPd in the optic nerve head region. Courtesy of A.M. Cantagallo [184].

(iii) the biomechanical behavior of sclera, choroid, retina, cornea and lamina cribrosa *via* an isotropic linear elasticity system.

We select 5 virtual patients, which input parameters are resumed in Tab. 16.1. We emphasize that

- all the patients have a systemic systolic/diastolic blood pressure of 120/80mmHg;
- baseline, P2 and P4 share similar TLPd, but this is a results of different IOP-RLTp situations;
- P1 and P3 have very different TLPd.

Parameter	Baseline	P1	P2	P3	P4	Unit
SP	120	120	120	120	120	mmHg
DP	80	80	80	80	80	mmHg
IOP	15	11	17	17	17	mmHg
RLTp	7	10	10	3	3	mmHg
TLPd	8	1	7	14	8	mmHg

Table 16.1 – Synthetic data of the patients who have been used in the TLPd virtual study.

2 Results

We present the results we have obtained by employing the OMVS in this virtual test case. Starting from the blood perfusion in the central retinal vessels, we cannot notice any remarkable difference for the CRA and CRV blood velocities in the prelaminar region (Fig. 16.2a); however Fig. 16.2b shows some fluctuations in the postlaminar CRV blood velocity profile for P2 and P3. These two patients share the same elevated IOP (17 mmHg), which may be responsible for these small variations.

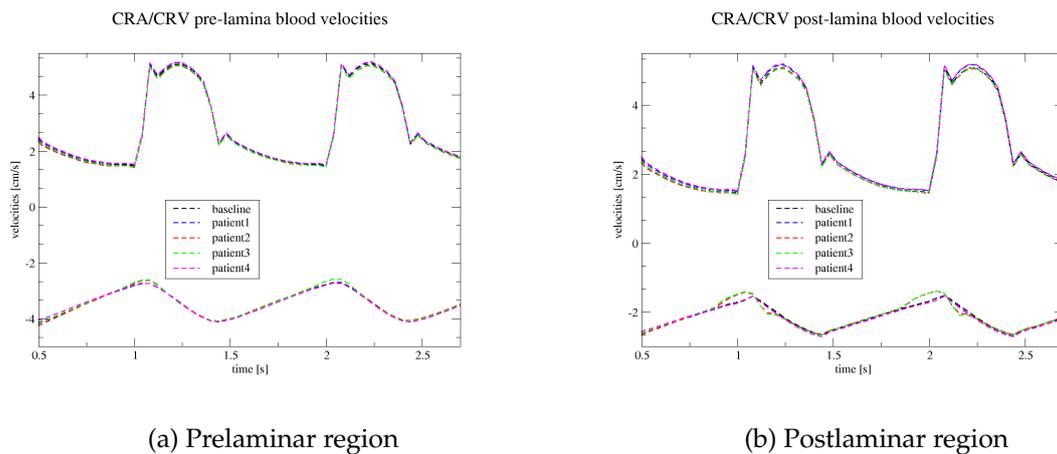


Figure 16.2 – CRA/CRV blood velocities comparison between the 5 different virtual subjects in the optic nerve head region.

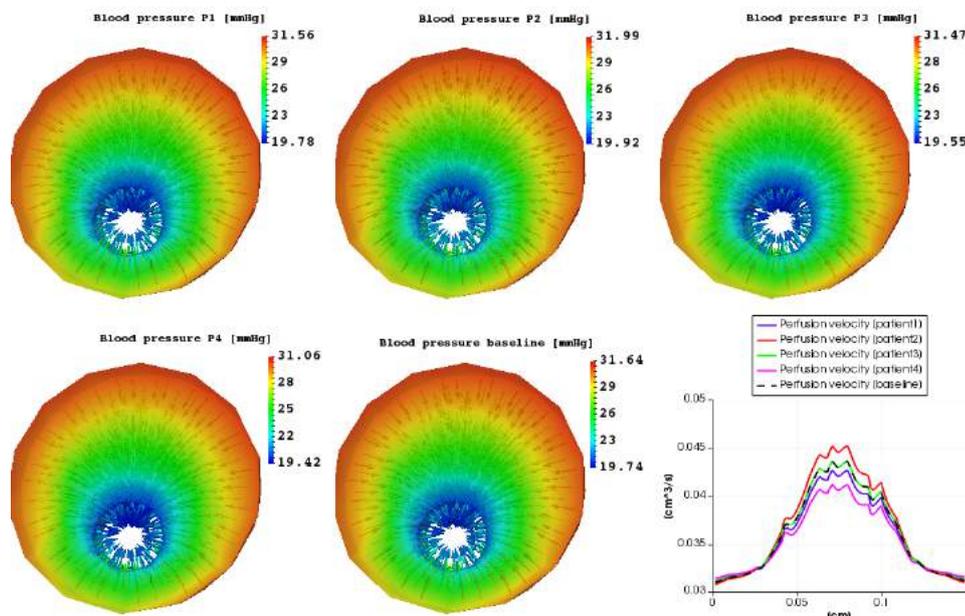


Figure 16.3 – Comparison of the hemodynamics in the lamina cribrosa between the 5 different virtual subjects.

Thanks to the OMVS we are able to predict the 3D spatial distribution of blood pressure and velocity within the lamina cribrosa (Fig. 16.3). As illustrated by the bottom right panel of Fig. 16.3, the model suggests that baseline, P2 and P4 have a perfusion difference within the lamina cribrosa up to 10%, even if they share similar TLPd. Moreover, Fig. 16.4 predicts a 1.4% discrepancy in the lamina cribrosa blood pressure gradient for baseline, P2 and P4.

The OMVS predicts also that the differences in the hemodynamics for these virtual patients don't have a biomechanical analogue. Indeed in Fig. 16.5 the lamina cribrosa displacements are similar for baseline, P2 and P4, whereas P1 and P3 exhibit an elevated discrepancy. This last fact is easily explained since P1 and P3 have very different TLPd, 1 and 14 mmHg, respectively.

Finally, for what concerns the other tissues, we cannot point out from the simulation results displayed in Fig. 16.6 that there are relevant differences in the 3D spatial distribution of the

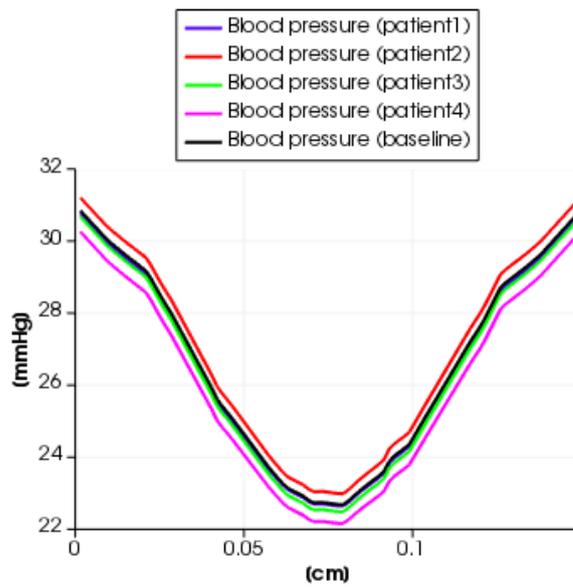


Figure 16.4 – Comparison of the lamina cribrosa blood pressure 1D cut along the diameter between the 5 different virtual subjects.

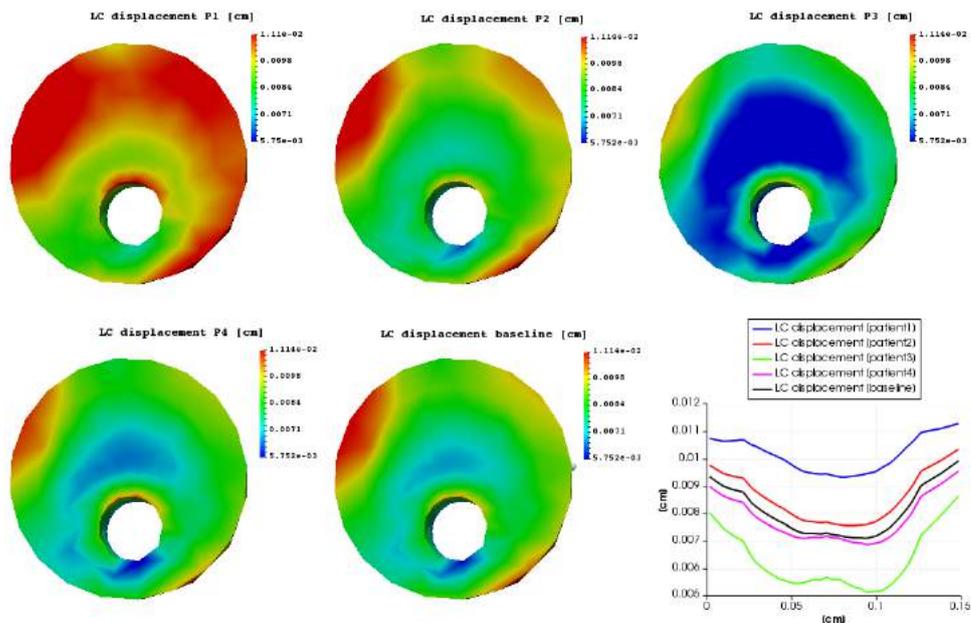


Figure 16.5 – Comparison of the biomechanics in the lamina cribrosa between the 5 different virtual subjects.

displacement among the 5 virtual patients. An explanation for this behavior is that the values chosen as input are not distributed enough to appreciate variations in the biomechanics of these tissues.

3 Conclusions

This virtual clinical test performed using the OMVS has shown that:

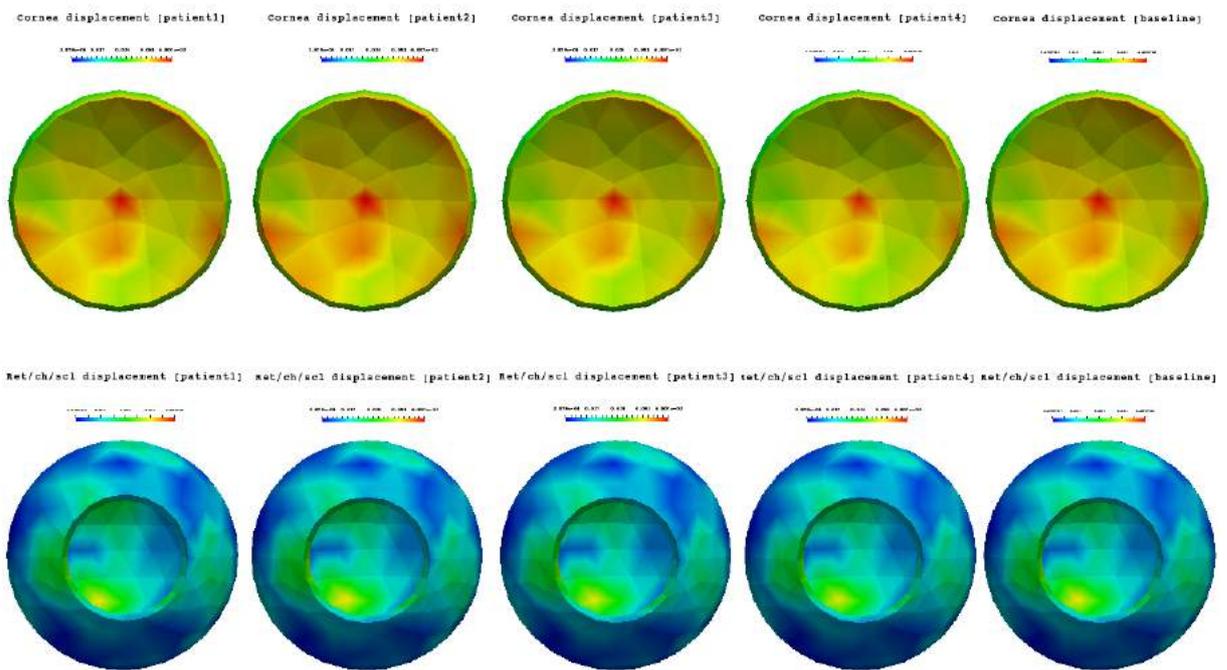


Figure 16.6 – Comparison of the biomechanics in the sclera, choroid, retina (bottom panels) and cornea (top panels) between the 5 different virtual subjects.

- elevated IOP influences the blood circulation in the postlaminal CRV;
- similar TLPd values may lead to similar biomechanics but different hemodynamics in the lamina cribrosa;
- different TLPd values may lead to similar hemodynamics but different biomechanics in the lamina cribrosa.

In conclusion, this model may also allow for novel findings in IOP regulation and RLTP effects in central retinal vein occlusion.

Contents

1	Origin of data	183
2	Data of 10 virtual patients	184
3	Simulation results	184
4	Conclusions	187

We present a patient database of synthetic data with an analysis of their simulation results.

1 Origin of data

The data that are used to generate the database are derived from experimental values reported in literature. We report the values and the sources we have adopted for our purpose in

Quantity	Symbol	Mean value	Std deviation	Unit	Source
Systolic blood pressure	SBP	116	23.2	mmHg	[254]
Diastolic blood pressure	DBP	69	13.8	mmHg	[254]
Heart rate	HR	69	15	bpm	[23]
Intraocular pressure	IOP	17	6	mmHg	[50]
Retrolaminar tissue pressure	RLTp	9.5	2.2	mmHg	[203]

Table 17.1 – Reference values for the virtual patients

Tab. 17.1.

2 Data of 10 virtual patients

Name	Gender	Age	SBP	DBP	HR	IOP	RLTp	Note
Tony	M	65	116.0	69.0	69	17	9.5	Baseline
John	M	47	116.0	69.0	69	26	9.5	High IOP
Tina	F	81	116.0	69.0	69	26	12.8	High IOP, RLTp
Margaret	F	55	150.8	89.7	69	26	12.8	High IOP, RLTp, BP
Sophie	F	59	116.0	69.0	69	17	12.8	High RLTp
Luke	M	61	116.0	69.0	69	17	6.2	Low RLTp
Max	M	57	116.0	69.0	69	11	9.5	Low IOP
Clara	F	69	116.0	69.0	69	11	6.2	Low IOP, RLTp
Jim	M	72	150.8	89.7	69	11	9.5	Low IOP, high BP
Jenny	F	85	116.0	69.0	100	17	9.5	High HR

Table 17.2 – Virtual patients data

The actual data for 10 virtual patients are reported in Table 17.2. The choices for the input have been made in order to span different cases that may be significant from an ophthalmological viewpoint.

We have selected a baseline subject (Tony), three glaucoma risk patients that have high IOP (John, Tina, Margaret), one subject with Idiopathic Intracranial Hypertension risk (Sophie), one subject with high translaminar tissue pressure difference due to intracranial hypotension (Luke), three patients with ocular hypotony risk (Max, Clara, Jim), and one patient with tachycardia (Jenny).

The goal of this patient selection is to vary few parameters at time in order to understand the influences of each parameter variation and disentangle the different effects.

The variation of each parameter has been based on experimental results briefly resumed in Tab. 17.1, in particular to simulate high conditions we have selected the mean value added to the standard deviation and for low values we chose mean minus the standard deviation. Only for IOP and HR we have made an exception, where the high values were pointed out directly from the literature [50, 23].

Undoubtedly this is just a preliminary analysis and it does not want to be exhaustive of any pathological situation, moreover we do not consider the fact that the parameters are not independent from each other. Despite these assumptions, we highlight the fact that the OMVS is still able to predict some clinically relevant conditions and to deal easily with clinical databases in view of more elaborated medical research.

3 Simulation results

In this section we illustrate the main results and outcomes that are obtained employing the OMVS on the virtual patient database described in the previous section and we sketch the discussion about the clinical impact and relevance of these simulations.

- **Tony** is an healthy subject that we have considered as the baseline. The profiles of the CRA and CRV blood flow agree with the profiles measured with medical instruments

such as the Eco-Doppler. Moreover the values obtained by the numerical simulations are in the physiological ranges [251]. The blood pressure and flow within the lamina cribrosa are in agreement with other work in literature [91]. All the other patients will be compared to his simulation results.

- **John** is a virtual patient that suffers from high IOP, therefore he is a glaucoma risk subject. The simulation results suggest a higher pressure and blood perfusion within the lamina cribrosa (up to 20%) and a lower blood flow (Fig. 17.1) both in the CRA (up to 66%) and in the CRV (up to 71%).

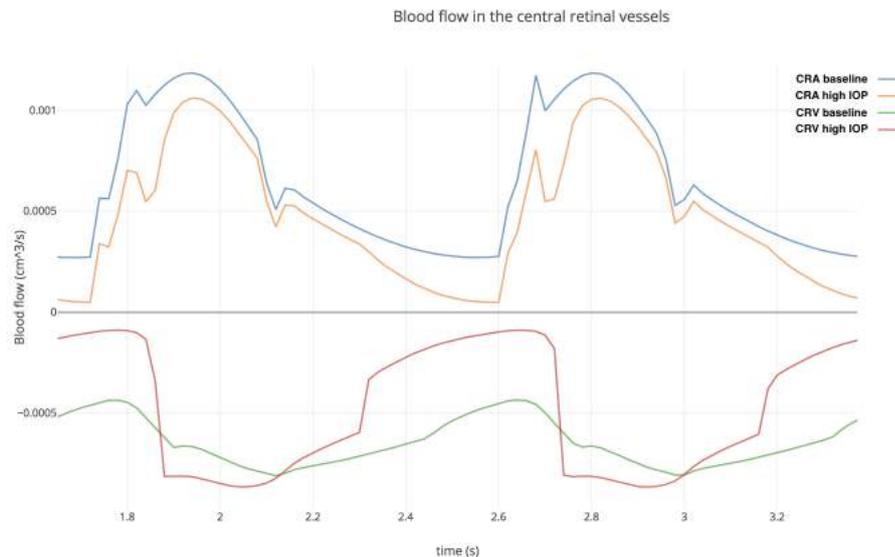


Figure 17.1 – Blood flow comparison between Tony and John using the Compare App.

- **Tina** is a virtual patient with both high IOP and RLTP that it is directly dependent from the intracranial pressure. This subject presents a slightly increase in the pressure and blood perfusion of the lamina cribrosa (up 9%), whereas she shows similar results than the previous patient (John, only high IOP) in the CRA and CRV (Fig. 17.2 lower blood flow 64% and 72%, respectively).
- **Margaret** is a virtual patient that suffers from high IOP, RLTP and blood pressure. The simulation results suggest higher blood pressure (up to 28.1%) and higher blood perfusion (28.2%) within the lamina cribrosa. The behavior of CRA and CRV is quite different (Fig. 17.3): CRA blood flow presents higher values up to 27%, whereas CRV blood flow profile is quite different from the baseline; indeed the twofold influence of high IOP and hypertension act on different segment of the CRV and the values attained are higher than baseline up to 58% at systolic peak or lower than baseline up to 43% at diastolic peak.
- **Sophie** is a virtual patient with a Idiopathic Intracranial Hypertension (IIH) risk. The simulation results do not significantly differ from the baseline values (Tony), in fact the lamina cribrosa hemodynamics is higher by 0.2%, the CRA flow is lower by 0.3%, and the CRV flow differs from the CRV baseline value by less than 0.1%.
- **Luke** is a virtual patient that suffers from intracranial hypotension. Also for this subject the simulation results are not particularly different from the baseline values. Lamina cribrosa blood pressure and blood perfusion differ from the healthy subject (Tony) by

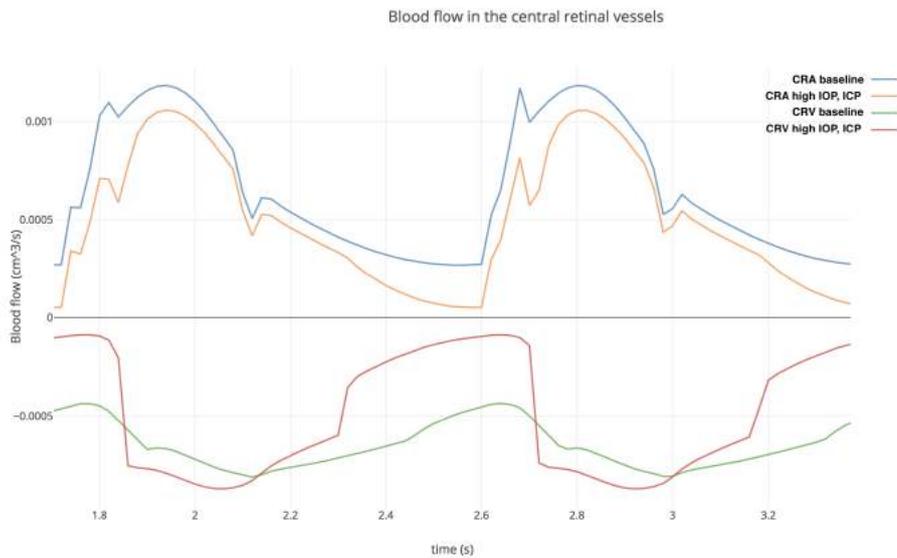


Figure 17.2 – Blood flow comparison between Tony and Tina using the Compare App.

0.2% and 0.4%, respectively. CRA and CRV blood flows are slightly higher (0.3%) than baseline results.

- **Max** is a virtual patient that has low IOP. The simulation results suggest a difference up to 2.2% for the blood pressure and up to 2.4% for the blood perfusion within the lamina cribrosa. CRA and CRV blood flows are higher than the baseline up to 4.5% and 13.9%, respectively.
- **Clara** is a virtual patient with ocular hypotony risk that suffers from low IOP and low RLTP. Lamina cribrosa hemodynamics is lower than baseline values up to 2%. CRA and CRV show a higher blood flow up to 4.7% and 14.2%, respectively.
- **Jim** is a virtual patient that suffers from ocular hypotony but also is an hypertensive patient. The simulation results suggest an increase in lamina cribrosa perfusion by 29.6% and an increase of CRA/CRV blood flow up to 39.2% and 35.4%, respectively.
- **Jenny** is a virtual patient with tachycardia. The blood pressure and flow profiles within lamina cribrosa, CRA and CRV reach similar peaks with respect to the baseline values, however the frequency of peaks and the shape of profiles is slightly different as expected.

In subjects with high IOP (John, Tina, Margaret), it is easy to point out a drop down of the blood flow especially in the CRV. This fact may be very likely due to the collapse of the veins, which is exactly what ophthalmologists observe from a physiological viewpoint. The CRV blood flow profile in Margaret is of particular interest: the blood pressure at diastolic is not enough to balance the high value of IOP of this subject. At a certain moment of each cardiac cycle the value of blood pressure is lower than the crucial value of external IOP and the blood flow drops down as depicted in Fig. 17.3; the vein collapses since the value of the internal pressure is lower than the external one.

As expected an increment of the blood pressure raises a proportional increase in the perfusion within the lamina cribrosa, CRA and CRV. For what concerns the IOP and the RLTP,

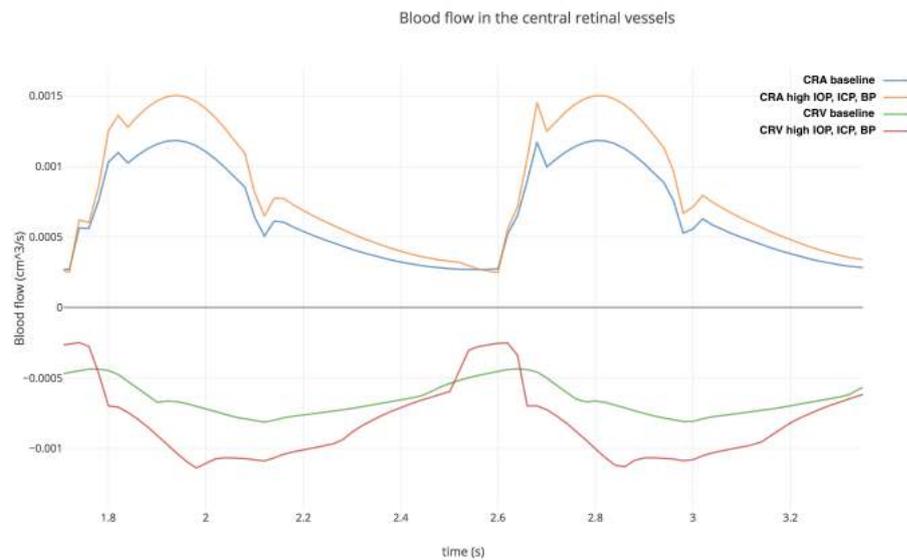


Figure 17.3 – Blood flow comparison between Tony and Margaret using the Compare App.

the simulation results suggest a direct proportionality in the lamina cribrosa whereas an inverse one is predicted in the central retinal vessels. This fact may be correlated to the need of blood to find new way of flowing in case of high IOP or RLTP, indeed an increment of these pressures leads to a greater external force on the vessels, which makes venules and veins collapse. However an increment of blood flow in the lamina cribrosa won't necessarily mean a better perfusion, rather an excessive blood flow rise may lead to a blow up in the optic disc microvasculature.

4 Conclusions

We have showed how the OMVS is able to simulate very different situations that are all significant from a clinical viewpoint. These results are easily accessible via the online visualization tool with the two different platforms we have implemented.

In conclusion, in this chapter we have presented an example of clinical research on simple virtual database, however a similar approach can be employed on a large dataset with real measurements.

Part VI

Sensitivity analysis

In the simulation result part (V) we have investigated different hemodynamics and biomechanics features varying some input in a deterministic way driven mainly by the questions of ophthalmologists. Since the deterministic results are promising, we understand the need of studying more deeply the variability of the model parameters and the propagation of errors from the input to the output, keeping in mind the discussion raised in Ch. 2 especially on the measurements. Thus, in this part, we present a sensitivity analysis study, in particular:

- a brief description of the methods we have employed to perform this study;
- a preliminary investigation to understand the propagation of uncertainties for the OMVS;
- the results of Sobol index study performed on *Level 0* of the OMVS.

Introduction to sensitivity analysis

Contents

1	Sobol index sensitivity analysis	194
2	Polynomial Chaos Expansion	196
3	Discussion on the IOP input parameter	198

Nowadays the interest on mathematical models applied to biomedical problems has greatly increased; in particular the need of a better understanding and knowledge of quantities in the medical context has raised tremendously the complexity of mathematics that is employed for the description of such physical systems. Thanks to its special connection to the brain and its accessibility to measurements, the eye provides a unique window on the brain, thereby offering non-invasive access to a large set of potential biomarkers that might help in the early diagnosis and clinical care of Neuro-Degenerative Diseases. We have developed an efficient computational framework to simulate and predict the mechanisms within the eye. In particular, our work focuses on the interaction between the hemodynamics of the main ocular vessels and the biomechanics of the principal ocular tissues.

One of the main challenges in medicine is to understand the influence that each internal or external factor has on the overall system; thanks to our mathematical model we are able to disentangle various effects, therefore providing a first insight of the impact of different components [210].

The great advantage of virtual mathematical and computational frameworks, also known as *in silico* models, is that experiments can be completed with large number of parameter sets with lower costs than gathering data from a patient cohort using *in vivo* models. Thus, the ophthalmologist may perform virtual exploratory wide researches in order to better design real experiments and population-based studies.

With these premises, we present a some results in the context of sensitivity analysis, which is study the propagation of uncertainties from in the input to the output of a mathematical model.

We have performed two analysis in order to investigate different aspects of the OMVS, in particular a preliminary investigation towards uncertainty quantification (Ch. 19) and a Sobol index study (Ch. 20).

In this chapter we give some ideas that are behind the Sobol index sensitivity (Sec. 1) and the polynomial chaos expansion technique (Sec. 2). For the complete theory and a detailed explanation we refer to [228] and [81, 17], respectively.

In the last section, we discuss an important aspect on the distribution of input parameter, in particular the IOP, trying to disentangle the scientific debate [142] about the more appropriate choice for the IOP probability density function.

1 Sobol index sensitivity analysis

If we consider the framework depicted by Fig. 18.1, we notice that the uncertainties in the input are propagated through the model to the output, however we have no information on this propagation and our level of confidence on the output. We write this relationship in a mathematical way:

$$\underline{y} = f(\underline{x}) \quad (18.1)$$

where \underline{x} is the vector of input, f is the model - in our case the OMVS - and \underline{y} is the vector of output we are interested in.

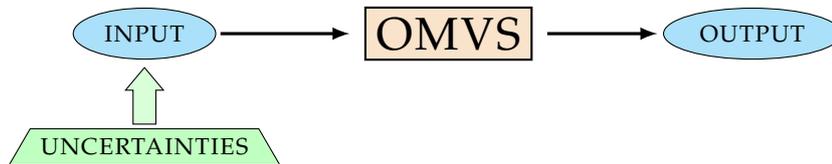


Figure 18.1 – Uncertainties propagation framework.

It is crucial for the Sobol index study that input and output are discrete finite information, for example the time profile of the blood flow in the CRA cannot be used, however its maximum, minimum or mean for each cardiac cycle is a suitable parameter for input or output.

The Sobol index study is very useful to understand the propagation of uncertainties mentioned above, however there are also some throwbacks that required particular attention: the high computational cost of the overall computation, the correlation between different input that it is not handled by the majority of algorithms and the possible interactions between different input within the model that lead to non-trivial high order effect on output.

If we consider d different input, we have that $\underline{x} = \{x_1, x_2, x_3, \dots, x_d\}$. For simplicity we keep just one output $\underline{y} = y$, but it is always possible to have a study with multiple output just performing independent sensitivity analysis.

All input x_i for $i = 1, \dots, d$ are independent and distributed following a probability density

function chosen by the user. Therefore we decompose $f(\underline{x})$:

$$y = f_0 + \sum_{i=1}^d f_i(x_i) + \sum_{i<j}^d f_{i,j}(x_i, x_j) + \cdots + f_{1,2,\dots,d}(x_1, x_2, \dots, x_d) \quad (18.2)$$

where f_0 is a constant, f_i is a function of x_i , $f_{i,j}$ is a function of x_i and x_j and so on. This functional decomposition has to be orthogonal, thus leads to the following definitions for condition expected values:

$$\begin{aligned} f_0 &= E[y] \\ f_i(x_i) &= E[y|x_i] - f_0 \\ f_{i,j}(x_i, x_j) &= E[y|x_i, x_j] - f_0 - f_i - f_j \end{aligned}$$

From these equations we highlight that if we consider the variation of just x_i we retrieve f_i , namely the *main effect* of input x_i on the output y , whereas if we vary x_i and x_j at the same time we obtain $f_{i,j}$ which is the effect of their individual variations added to their interactions, namely the *second-order interaction*. We can iterate this procedure to define all the higher order terms.

Let us assume $f(\underline{x})$ square-integrable, (18.2) leads to the decomposition of the variance:

$$\sigma_y^2 = \sum_{i=1}^d V_i + \sum_{i<j}^d V_{i,j} + \cdots + V_{1,2,\dots,d} \quad (18.3)$$

where

$$\begin{aligned} V_i &= \sigma_{x_i}^2 [E_{\underline{x}\setminus i}[y|x_i]] \\ V_{i,j} &= \sigma_{x_{i,j}}^2 [E_{\underline{x}\setminus i,j}[y|x_i, x_j]] - V_i - V_j \\ &\dots \end{aligned}$$

$\underline{x}\setminus i$ means all set of variables in \underline{x} except x_i .

Thanks to (18.3) we define the Sobol index also called sensitivity index. The *main effect index* or *first-order sensitivity index* is

$$S_i := \frac{V_i}{\sigma_y^2} \quad (18.4)$$

and has the property that

$$\sum_{i=1}^d S_i + \sum_{i<j}^d S_{i,j} + \cdots + S_{1,2,\dots,d} = 1$$

where $S_{i,j} = \frac{V_{i,j}}{\sigma_y^2}$ and $S_{1,2,\dots,d} = \frac{V_{1,2,\dots,d}}{\sigma_y^2}$ are the high-order sensitivity indices defined similarly to (18.4).

One of the main purposes, however, is to quantify the importance of each variable on the output variance and not just the effect of varying one input alone, averaged over variations in other input parameters. The problem is that this computation requires the evaluation of $2^d - 1$

indices, which is too expensive.

The *total-effect index* or the *total-order index* S_{T_i} is the solution since it measures the contribution of the output variance of x_i including all the variances caused by its interactions, of any order, with any other input variables:

$$S_{T_i} = \frac{E_{\underline{x} \setminus i} [\sigma_{x_i}^2 [y | \underline{x} \setminus i]]}{\sigma_y^2} = 1 - \frac{\sigma_{\underline{x} \setminus i}^2 [E_{x_i} [y | \underline{x} \setminus i]]}{\sigma_y^2} \quad (18.5)$$

In contrast to first-order, $\sum_{i=1}^d S_{T_i} \geq 1$ by reason of the interaction effects that are counted multiple times. The sum will be exactly equal to 1 only if the model is purely additive.

2 Polynomial Chaos Expansion

The idea of polynomial chaos (PC) expansion is to rewrite Eq. (18.1) using a polynomial basis, in particular:

$$y = \sum_{k=1}^{\infty} \beta_k \psi_k(\underline{x}) \quad (18.6)$$

where $\psi_k(\underline{x})$ for $k \in \mathbb{N}$ are multivariate polynomials that are orthonormal with respect to the joint probability density function $h_{\underline{x}}(\underline{x})$, which is:

$$\langle \psi_j(\underline{x}), \psi_k(\underline{x}) \rangle \equiv E [\psi_j(\underline{x}) \psi_k(\underline{x})] = \delta_{j,k}$$

where $\delta_{j,k} = 1$ if $j = k$ and 0 otherwise, and β_k for $k \in \mathbb{N}$ are deterministic coefficients that fully characterize the output y .

The problem that has to be faced now is how to build the PC basis.

In a first approach we consider that \underline{x} is a vector of d **independent random variables**. Basically we can rescale x_i using a specific mapping T_i :

$$U_i = T_i(x_i) \quad i = 1, \dots, d.$$

For simplicity we assume that $U_i = x_i$ and the choices for these U_i are standard distributions, e.g. normal, uniform, ...

If the variables are independent we can rewrite the joint probability density function as

$$h_{\underline{x}}(\underline{x}) = \prod_{i=1}^d h_{x_i}(x_i)$$

where $h_{x_i}(x_i)$ is the marginal probability function of x_i .

Considering a family of orthonormal polynomials $\pi_j^{(i)}$ for $j \in \mathbb{N}$, which depends on the distribution of x_i (see Tab. 18.1 for some examples), we have that:

$$\langle \pi_j^{(i)}(x_i), \pi_k^{(i)}(x_i) \rangle \equiv E [\pi_j^{(i)}(x_i) \pi_k^{(i)}(x_i)] = \delta_{j,k}$$

Distribution	Polynomial family
Uniform	Legendre
Normal	Hermite
Gamma	Laguerre
Beta	Jacobi
Binomial	Krawtchouk
Poisson	Charlier
Lognormal	Stieltjes-Wigert

Table 18.1 – Examples of orthogonal polynomials associated with the random variables with a specific probability density function.

Therefore we obtain the set of orthonormal multivariate polynomial $\{\psi_{\underline{\alpha}}(\underline{x}), \underline{\alpha} \in \mathbb{N}^d\}$ tensorizing the d resulting families of univariate polynomials:

$$\psi_{\underline{\alpha}}(\underline{x}) = \pi_{\alpha_1}^{(1)}(x_1) \times \cdots \times \pi_{\alpha_d}^{(d)}(x_d) \quad (18.7)$$

where $\alpha = \{\alpha_1, \dots, \alpha_d\}$.

In case of **dependent random variables** we can generate an orthonormal basis with

$$\psi_{\underline{\alpha}}(\underline{x}) = \tilde{K}(\underline{x}) \prod_{i=1}^d \pi_{\alpha_i}^{(i)}(x_i) \quad (18.8)$$

where $\tilde{K}(\underline{x})$ is a function of the copula of \underline{x} .

Once our full model $f_{\underline{x}}$ has been expanded using a PC basis (Eq. (18.6)), we can build the **metamodel** that is the finite representation of the full model using a finite subset of basis:

$$\hat{y} = \hat{f}_{\underline{x}}(\underline{x}) = \sum_{k=1}^P \beta_k \psi_k(\underline{x}) \quad (18.9)$$

where P is defined as follow $P + 1 = \frac{(d + N_0)!}{d!N_0!}$ and N_0 is the polynomial order where we truncate the expansion.

The great advantage of using PC expansion is that we obtain easily the Sobol indices. In fact both PC and Sobol expansions are sums of orthogonal functions, in other words, from a given PC expansion, the Sobol indices at any order may be obtained by a combination of the squares of the coefficients $\hat{\beta}_k$.

Following the works of [233, 53] we have to define two sets of indices needed for first-order and total-order Sobol indices:

$$\mathbb{J}_{i_1, i_2, \dots, i_s} = \begin{cases} \underline{\alpha} : \alpha_k > 0 \ \forall k = 1, \dots, d \ k \in (i_1, i_2, \dots, i_s) \\ \underline{\alpha} : \alpha_k = 0 \ \forall k = 1, \dots, d \ k \notin (i_1, i_2, \dots, i_s) \end{cases} \quad (18.10)$$

$$\mathcal{J}_{i_1, i_2, \dots, i_s} = \{\underline{\alpha} : \alpha_k > 0 \ \forall k = 1, \dots, d \ k \in (i_1, i_2, \dots, i_s)\} \quad (18.11)$$

In other words Eq. (18.10) tells that \mathbb{J}_i picks the single variables i , $\mathbb{J}_{i,j}$ picks the pair of variables (i, j) and so on, whereas Eq. (18.11) represents a set of valid indices that selects all the

terms in the PC expansion where the variable x_i is present.

Thus, the first-order index

$$\hat{S}_i = \frac{\sum_{j \in \mathbb{J}_i} \hat{\beta}_j^2}{\sum_{k=1}^P \hat{\beta}_k^2} \quad (18.12)$$

and the total-order index

$$\hat{S}_i^{tot} = \frac{\sum_{j \in \mathcal{J}_i} \hat{\beta}_j^2}{\sum_{k=1}^P \hat{\beta}_k^2} \quad (18.13)$$

can be analytically computed from the coefficients of the PC expansion at a negligible computational cost.

This technique leads to a remarkable gain with respect to other methods based on Monte Carlo simulation [114].

3 Discussion on the IOP input parameter

In literature there is a notable discussion about normal and lognormal probability density functions and which of the two can better represent biological phenomena [142].

In view of our application we consider as example the IOP distribution with the data recovered from [50] (mean $\mu_{normal} = 17\text{mmHg}$, standard deviation $\sigma_{normal} = 6\text{mmHg}$). Starting from these values that refers to a Gaussian distribution, we have computed the mean and the variance also for a lognormal behavior using the following formulas [120, Ch. 14: Lognormal Distributions]:

$$\mu_{lognormal} = \log \left(\frac{\mu_{normal}}{\sqrt{1 + \frac{\sigma_{normal}^2}{\mu_{normal}^2}}} \right) \quad \sigma_{lognormal}^2 = \log \left(1 + \frac{\sigma_{normal}^2}{\mu_{normal}^2} \right)$$

In this context we performed a comparative analysis using three different IOP distribution based on the same clinical data, which values has been defined above. In particular, we compare a *normal*, a *truncated normal* and a *lognormal* distribution.

Fig. 18.2 highlights that the normal probability density (blue) is going beyond some physiological constraints for healthy patients such as $IOP > 5\text{mmHg}$. The truncated normal probability density function (green) does not show this issue, however this IOP distribution presents an abnormal cut in the left tail, which makes it ineffective for a sensitivity analysis (such as the one discussed in Ch. 20). In our simulation with the data provided by Colton et al.[50], the lognormal distribution seems to be the more natural one. This possibility has been accepted also in other works [246, 234].

Since the use of one distribution with respect to the other may depend largely by the input data employed, - for instance for some couple of values of mean/standard deviation the normal distribution may not generate unphysiological situation - we used both distributions, normal and lognormal, to sample the IOP in the following chapters. Specifically, in Ch. 19 we use the normal probability density function to perform a propagation of uncertainties study. In this case the interest of this investigation is to understand the correlation between input and output

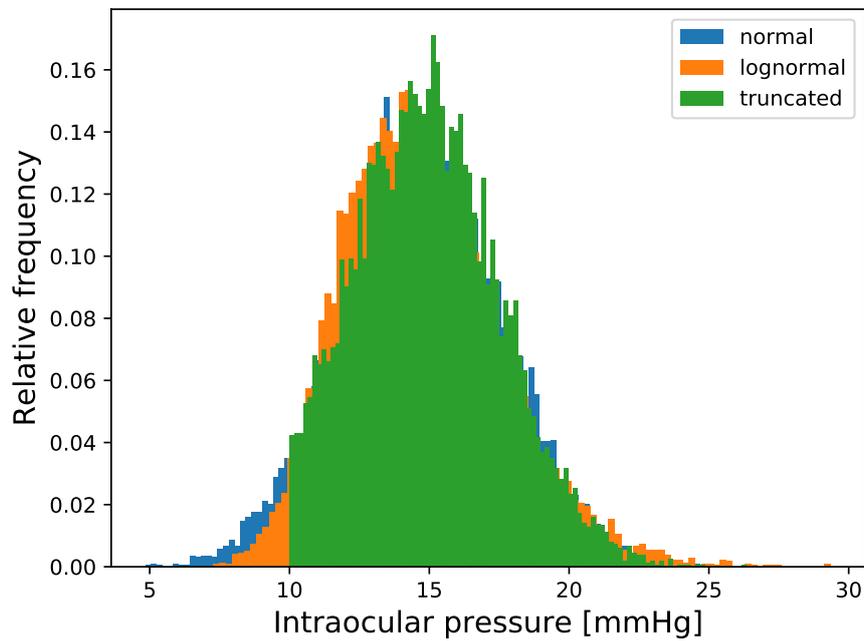


Figure 18.2 – Comparison among normal (blue), lognormal (orange) and truncated normal (green) IOP distributions using the same input data [50].

on a real clinical database, therefore we considered the non-physiological low IOP values as possible errors that can rise in collecting the data. On the other hand, Ch. 20 is focused on a sensitivity analysis using Sobol indices, thus we would like to avoid any miscalculation due to unphysiological values that may lead to unrealistic discontinuities in the results. For this reason we utilized the lognormal distribution based on the population based study operated by Suh et al. [234].

Preliminary uncertainty propagation analysis

Contents

1	Data	201
2	Implementation	202
3	Results	204
4	Conclusions	205

The first sensitivity analysis is a preliminary investigation in order to understand the propagation of uncertainties for *Level 0* of the OMVS (Ch. 3).

First, we present the Python code we have implemented to perform this analysis, then we specify the data used and the results achieved. The idea of this analysis is to simulate the mathematical model many times varying some parameters and look at the distribution of the output.

1 Data

We have completed three different sets of $N = 60000$ evaluations of the model varying the IOP. We recall that the IOP has a crucial role in the description of the vein collapsibility, which is modeled in the OMVS by Starling resistors (see Eq. (3.8)); namely, when the external pressure is higher than the internal blood pressure, veins collapse, therefore dropping down the blood flow.

We have sampled the IOP with a normal distribution that has mean equal to 17 mmHg and standard deviation of 6 mmHg [50].

In this study, we have selected three significant cases where the three sets of evaluations differ from each other by the blood pressure value imposed [91]:

- *baseline* with a systolic/diastolic blood pressure (SP/DP) of 120/80 mmHg;
- *low* with SP= 100 mmHg and DP= 70 mmHg ;
- *high* that is initialized with SP/DP = 140/90 mmHg.

2 Implementation

In order to carry out this study, we have implemented the analysis thanks to the statistical library OpenTURNS [10] created in Python.

First we have exported our model developed in OpenModelica to FMU format (Sec. 2, Ch. 10). This allows us to use the PyFmi Python library [7] in order to interact with the FMU version of the OMVS and simulate it.

Code 19.1 – Model definition

```

1 class model_compute(OpenTURNSPythonFunction):
2     def __init__(self, names, modelFMU):
3         self.names = names;
4         self.modelFMU = modelFMU;
5         OpenTURNSPythonFunction.__init__(self, n_inputs, n_outputs)
6     def _exec(self, X):
7         modelFMU.reset()
8         global iterator
9         n_var=n_inputs
10        if names[0] == 'IOP':
11            IOP = X[0];
12            DP = 80
13            SP = 120
14            RLTP = 7
15            #
16            modelFMU.set('Pin.SP', SP);
17            modelFMU.set('Pin.DP', DP);
18            modelFMU.set('IOP_source.V', IOP);
19            modelFMU.set('RLTp_source.V', RLTP);
20            #
21            opts = modelFMU.simulate_options()
22            opts['filter'] = ['time', 'lcR.v', 'lcR.p.i', 'R1a.p.i', 'R5a.n.i', 'Pin.HR']
23            opts['result_file_name'] = 'baseline_res.txt'
24            #
25            res = modelFMU.simulate(final_time=8., options=opts)
26            # recover results
27            time = res['time']
28            p_lamina = -res['lcR.v']
29            j_lamina = -res['lcR.p.i']
30            j_CRA = -res['R1a.p.i']
31            j_CRV = -res['R5a.n.i']
32            T_c = 60/res['Pin.HR'];
33            sel_it = np.where(time > 6*T_c)
34
35            maxCRV = max(j_CRV[sel_it]);
36            maxCRA = max(j_CRA[sel_it]);
37            maxLC = max(j_lamina[sel_it]);
38            print('\n#####')
39            print('Simulation %d completed.' % iterator)
40            print('#####\n')
41            iterator = iterator + 1;
42            return [maxCRV, maxCRA, maxLC];

```

The Python code reported in 19.1 describes the class that computes a single simulation, in particular we:

- i. reset the model \Rightarrow line 7;

- ii. initialize the input parameters of the model, in this case the only parameter that is varying for each simulation is the IOP that is taken from its distribution that has been defined outside and passed as $X \Rightarrow$ lines 10 – 14;
- iii. set these parameters in the FMU model thanks to the PyFmi method `set` \Rightarrow lines 16 – 19;
- iv. define the simulation options, in particular we export just the quantities of interest \Rightarrow line 22;
- v. simulate the model \Rightarrow line 25;
- vi. recover all the significant results \Rightarrow line 27 – 32;
- vii. compute the output, in particular we are interested in the maximum blood flow in the CRA, CRV and lamina cribrosa starting from the sixth cardiac cycle when the system has already reached periodic state \Rightarrow lines 33 – 37.

Code 19.2 – Initialization

```

1 iterator = 1;
2 names = ['IOP'];
3 output_names = ['CRV', 'CRA', 'LC'];
4 n_inputs = len(names);
5 n_outputs = len(output_names);
6 size = 20000
7 doSecondOrder = False;
8
9 ## load model
10 curr_dir = O.path.dirname(O.path.abspath(__file__));
11 fmu_name = O.path.join(curr_dir, 'retina.fmu')
12 modelFMU = load_fmu(fmu_name)
13
14 ##### Set distributions
15 IOP_mean = 17; IOP_dev = 6;
16 dist_IOP = Normal(IOP_mean, IOP_dev);
17
18 comptemp = Description(1)
19 inpCollection = DistributionCollection(n_inputs);
20 #
21 disttemp = Distribution(dist_IOP)
22 comptemp[0] = "IOP";
23 disttemp.setDescription(comptemp)
24 disttemp.setName("random variable IOP")
25 inpCollection[0] = disttemp

```

In Code 19.2 we show how we create some useful parameters (lines 1 – 7), load the FMU model (lines 10 – 12) and set the IOP as a normal distribution (lines 15 – 25).

Code 19.3 – Simulation

```

1 CorrMat = CorrelationMatrix(inpCollection.getSize())
2 aCopula = IndependentCopula(inpCollection.getSize())
3 aCopula.setName("Copula of the random input vector")
4 myDist = ComposedDistribution(inpCollection, Copula(aCopula))
5 inp = RandomVector(Distribution(myDist))
6
7 model = Function(model_compute(names, modelFMU))
8
9 RandomGenerator.SetSeed(0)
10
11 inputDesign = SobolIndicesExperiment(myDist, size, doSecondOrder).generate()
12 outputDesign = model(inputDesign)
13
14 np.save("Figures/baseline-outputData.dat", outputDesign)

```

Code 19.3 exploits some OpenTurns method to solve the FMU model using the distribution initialized before as input for each simulation, in particular in line 11 we use the Sobol index analysis method to set the number of experiments and the samples that are therefore solved in line 12 and then the results saved (line 14).

3 Results

We mainly focus on the following output computed by the python code described by the previous section:

- i) *CRA*: the simulated maximum blood flow within the CRA;
- ii) *CRV*: the simulated maximum blood flow within the CRV;
- iii) *LC*: the simulated maximum blood flow within the lamina cribrosa.

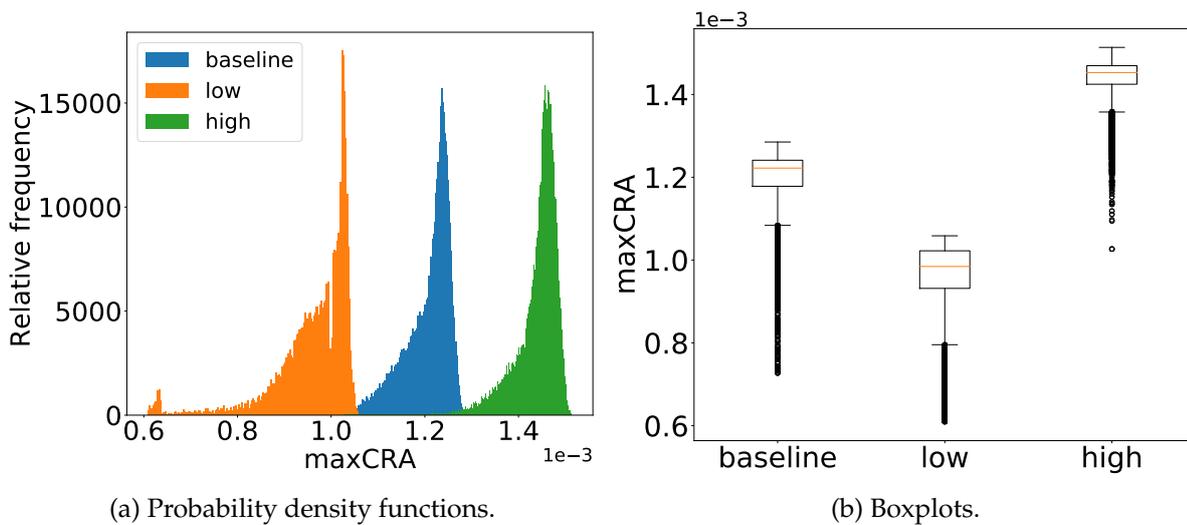


Figure 19.1 – Comparison of *CRA* blood flows for preliminary uncertainty quantification analysis.

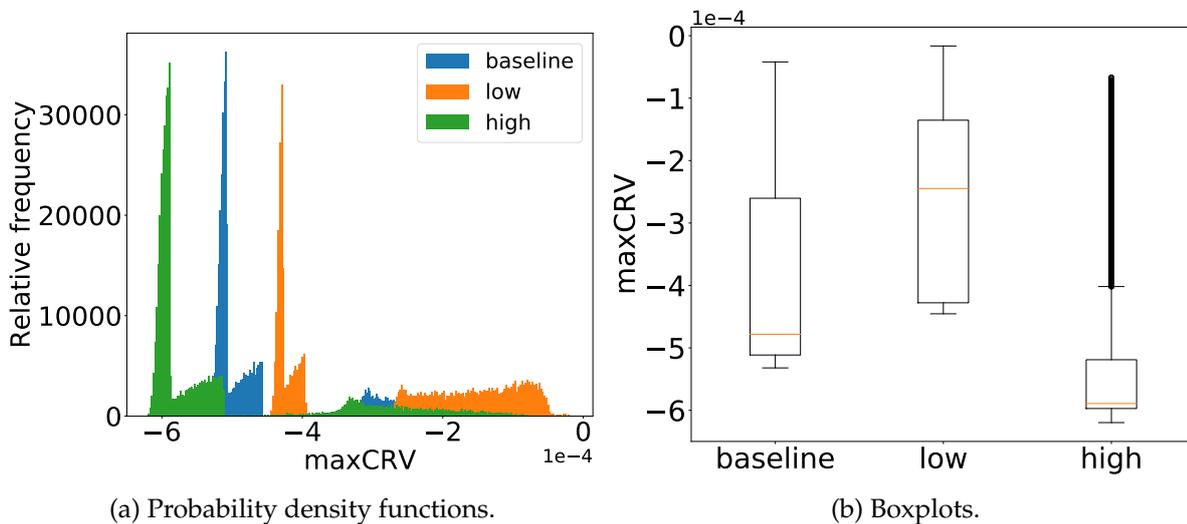


Figure 19.2 – Comparison of *CRV* blood flows for preliminary uncertainty quantification analysis..

As expected and illustrated by Figs. 19.1, 19.2 and 19.3, the initial blood pressure is influencing the distribution of the outputs varying the IOP. Thus, we highlight that:

- the predicted *CRA* blood flow median value attains $0.0014529 \text{ cm}^3/\text{s}$, which is very close to experimental mean value reported by Dorner et al. [62] ($0.00131463636 \text{ cm}^3/\text{s}$);

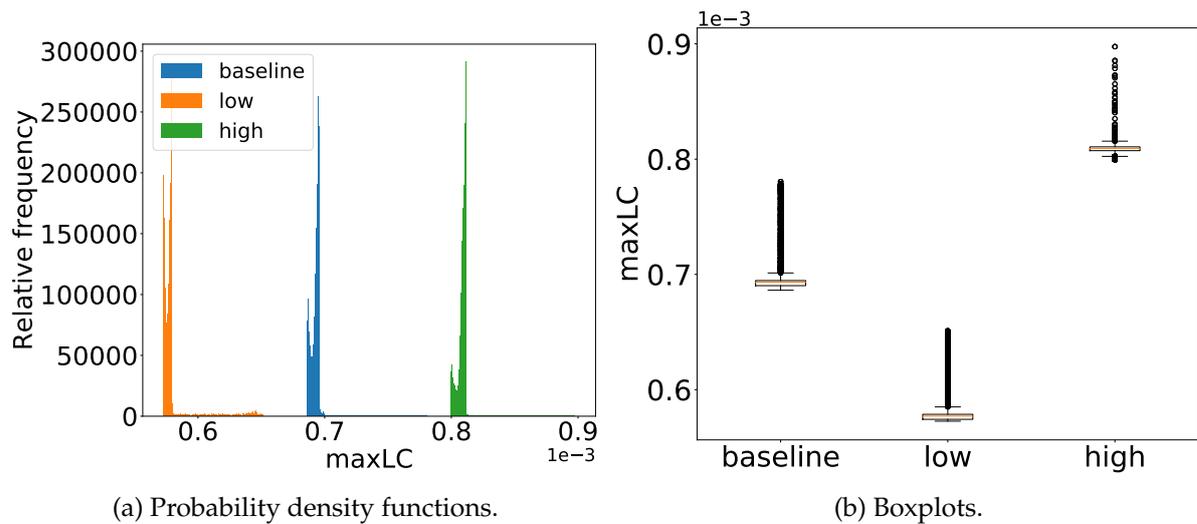


Figure 19.3 – Comparison of *LC* blood flows for preliminary uncertainty quantification analysis..

- there are other factors that significantly influence the lamina cribrosa hemodynamics; indeed its computed blood flow variability is considerably lower than the other two cases as shown by Figs. 19.3;
- the CRV blood flow probability density function (Fig. 19.2a) exhibits two different behaviors: a peak of frequencies with quasi-Gaussian distribution for high values and a frequency plateau for low values separated by an almost empty frequency area of middle values. Moreover it suggests a different repartition of frequencies between the three cases, in particular, for low blood pressures, the high values peak is narrower and the plateau is wider and with a major number of results than the high blood pressure case. This fact is confirmed by the boxplots in Fig. 19.2b, indeed the tail of CRV blood flow low values is within the first and third quantile range for the low blood pressure case, whereas this tail is composed just by outliers for high blood pressures.

This last finding may lead an interesting physiological interpretation; as reminded in Sec. 1, the IOP has a non-linear effect on the retinal vasculature, in particular on the venous part. Following this reason, the CRV high values peak may represent the natural state when the IOP is lower than the venous blood pressure, whereas the low values plateau denotes the collapse state. This statement is consistent with the previous analysis where we found a wide plateau for low blood pressure and a important frequency in the peak for high blood pressures.

4 Conclusions

The proposed preliminary study presents a twofold interest: mathematical and clinical. From a bioengineering viewpoint, we have set up the framework for an uncertainties quantification analysis, which allows us to evaluate the effects of the propagation of uncertainty from input to output. From a medical viewpoint, we have proposed a computational tool that may serve as a complementary instrument to the clinical and experimental research.

In the suggested virtual experiment, we have combined (i) a deterministic model, which is

based on biophysical laws, and (ii) a stochastic input that depends on experiments, in order to understand the variability of the ocular posterior tissue hemodynamics.

Sobol index sensitivity analysis

Contents

1	Data	207
2	Implementation	208
3	Results	212
4	Conclusions	215

In the second sensitivity analysis we use as mathematical model the *Level 0* of the OMVS (Ch. 3) as in the previous chapter, however in this case we perform a sensitivity analysis using Sobol indices (see Sec. 1, Ch. 18). In particular we present the implementation of this study using the polynomial chaos expansion method (see Sec. 2, Ch. 18) *via* Python and the results obtained.

1 Data

In this study we have as input random variables the IOP, the RLTP, the SP and DP in order to compute the Sobol indices.

In the model, IOP and RLTP act in the description of the collapsibility of veins due to external mechanical pressures, which is modeled by Starling resistors (see Eq. (3.8)), whereas SP and DP are essential for the reconstruction of the input blood flow (see Eq. (3.9)).

For the data used in this analysis we have taken from [234] the mean value of IOP equal to 14.7mmHg and a standard deviation value of 2.8mmHg. The lognormal probability density function used as input for the IOP in the study is shown in Fig. 20.1a.

For the RLTP, SP and DP we have used normal distributions and we refer to the Figs. 20.1b,20.1c and 20.1d, respectively, for their probability density functions, and to Tab. 20.1 for the values adopted in the simulations.

Input variable	Mean [mmHg]	Std deviation [mmHg]	Reference
RLTp	9.5	2.2	[203]
SP	124.1	11.1	[218]
DP	77.5	7.1	[218]

Table 20.1 – Input data distribution for the Sobol index study

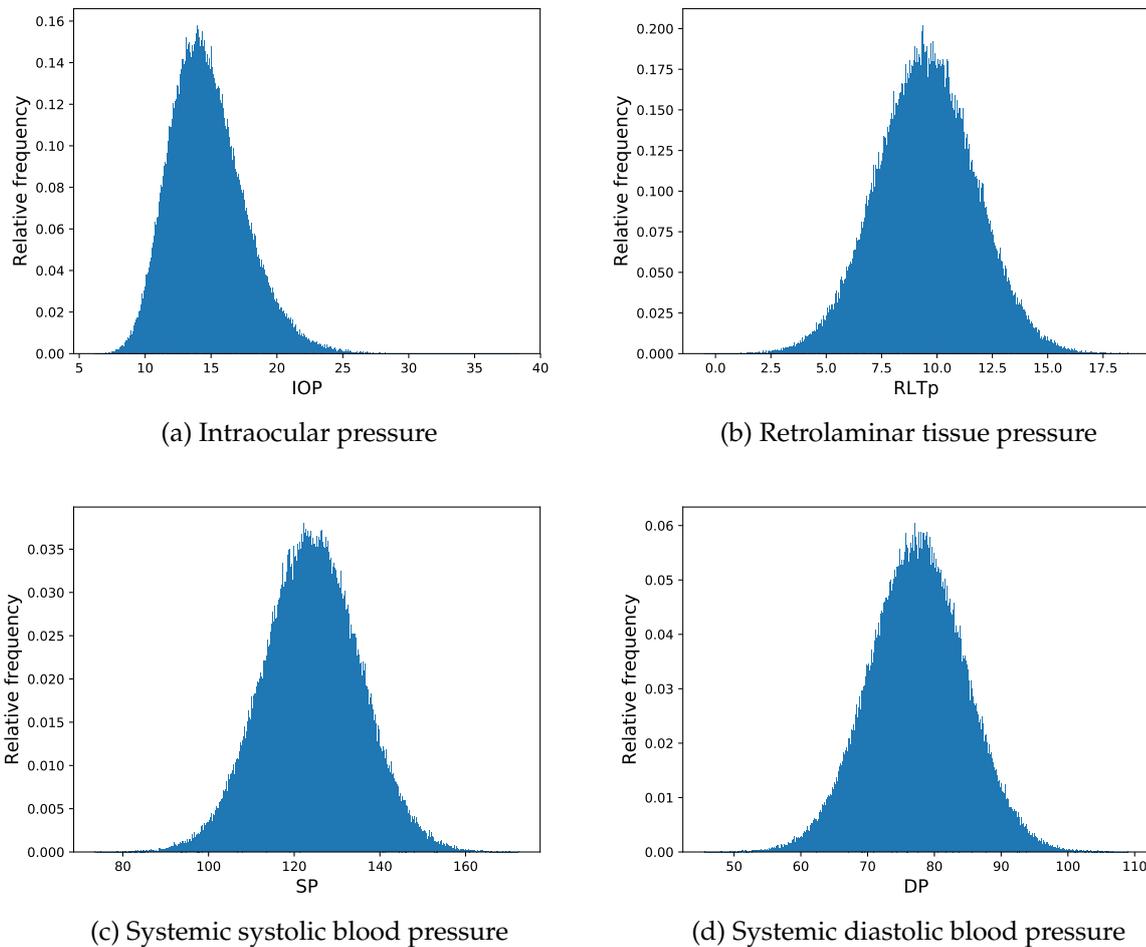


Figure 20.1 – Probability density functions of the input variables for the Sobol index study.

2 Implementation

As for the previous chapter we have implemented this study in Python with the help, in particular, of the two libraries PyFmi [7] and OpenTurns [10].

Code 20.1 – Model definition

```

1 class model_compute(OpenTURNPythonFunction):
2     def __init__(self, names):

```

```

3         self.names = names;
4         self.modelFMU = modelFMU;
5         OpenTURNPythonFunction.__init__(self, n_inputs, n_outputs)
6     def _exec(self,X):
7         global iterator
8         modelFMU.reset();
9         n_var=n_inputs
10        if names[0] == 'IOP':
11            IOP = X[0];
12        if names[1] == 'RLTp':
13            RLTP = X[1];
14        if names[2] == 'SP':
15            SP = X[2];
16        if names[3] == 'DP':
17            DP = X[3];
18        #
19        modelFMU.set('Pin.SP',SP);
20        modelFMU.set('Pin.DP',DP);
21        modelFMU.set('IOP_source.V',IOP);
22        modelFMU.set('RLTp_source.V',RLTP);
23        #
24        opts = modelFMU.simulate_options()
25        opts['filter'] = ['time','lcR.v','lcR.p.i','R1a.p.i','R5a.n.i','Pin.HR']
26        #
27        res = modelFMU.simulate(final_time=8.,options=opts)
28        # recover results
29        time = res['time']
30        p_lamina = -res['lcR.v']
31        j_lamina = -res['lcR.p.i']
32        j_CRA = -res['R1a.p.i']
33        j_CRV = res['R5a.n.i']
34        T_c = 60/res['Pin.HR'];
35        sel_it = np.where(time > 7*T_c)
36
37        maxCRV = max(j_CRV[sel_it]);
38        maxCRA = max(j_CRA[sel_it]);
39        maxLC = max(j_lamina[sel_it]);
40        print('\n#####')
41        print('Simulation %d/%d completed.' % (iterator,total_n_sim) )
42        print('#####\n')
43        iterator = iterator + 1;
44        return [minCRV, maxCRA, maxLC];

```

The Python code reported in 20.1 describes the class that computes a single simulation, in particular we:

- i. reset the model \Rightarrow line 8;
- ii. initialize the input parameters of the model, in particular IOP, RLTP, SP and DP \Rightarrow lines 10 – 17;
- iii. set these parameters in the FMU model thanks to the PyFmi method set \Rightarrow lines 19 – 22;
- iv. define the simulation options, in particular we export just the quantities of interest \Rightarrow line 25;
- v. simulate the model \Rightarrow line 27;
- vi. recover all the significant results \Rightarrow line 29 – 34;
- vii. compute the output, in particular we are interested in the maximum blood flow in the CRA, CRV and lamina cribrosa starting from the seventh cardiac cycle when the system has already reached periodic state \Rightarrow lines 35 – 39.

Code 20.2 – Initialization

```

1     iterator = 1;
2     names = ['IOP','RLTp','SP','DP'];
3     output_names = ['minCRV', 'maxCRA', 'maxLC'];
4     n_inputs = len(names);
5     n_outputs = len(output_names);
6     size = 200000
7     size_testPC = 500
8     sampleCreatePC = 1500
9     total_n_sim = size_testPC + sampleCreatePC;
10
11     # load model
12     curr_dir = O.path.dirname(O.path.abspath(__file__));
13     fmu_name = O.path.join(curr_dir,'../retina.fmu')
14     modelFMU = load_fmu(fmu_name)
15
16     ##### Set distributions
17     #

```

```

18 IOP_mean = 14.7; IOP_dev = 2.8;
19 IOP_var = IOP_dev*IOP_dev;
20 IOP_mu = np.log(IOP_mean/np.sqrt(1+IOP_var/(IOP_mean*IOP_mean)))
21 IOP_sigma = np.sqrt(np.log(1+IOP_var/(IOP_mean*IOP_mean)))
22 #
23 RLTP_mean = 9.5; RLTP_dev = 2.2
24 #
25 SP_mean = 124.1; SP_dev = 11.1;
26 SP_var = SP_dev*SP_dev;
27 #
28 DP_mean = 77.5; DP_dev = 7.1;
29 DP_var = DP_dev*DP_dev;
30 ###
31 dist_IOP = LogNormal(IOP_mu, IOP_sigma );
32 #
33 dist_RLTP = Normal(RLTP_mean, RLTP_dev);;
34 #
35 dist_SP = Normal(SP_mean, SP_dev);
36 dist_DP = Normal(DP_mean, DP_dev);

```

Code 20.2 displays:

- i. the creation of some useful parameters \Rightarrow lines 1 – 5;
- ii. the choice of the sizes for (line 6) the Sobol index analysis using the PC expansion, (line 7) test to check the validity of the PC and (line 8) the number of full model simulation needed to build the PC;
- iii. the FMU model loading \Rightarrow lines 12 – 14;
- iv. the set of the IOP distribution \Rightarrow lines 18 – 21 and 31;
- v. the set of the RLTP distribution \Rightarrow lines 23 and 33;
- vi. the set of the SP distribution \Rightarrow lines 25 – 26 and 35;
- vii. the set of the DP distribution \Rightarrow lines 28 – 29 and 36;

Code 20.3 – Polynomial Chaos definition

```

1 CorrMat = CorrelationMatrix(inpCollection.getSize())
2 aCopula = IndependentCopula(inpCollection.getSize())
3 aCopula.setName("Copula of the random input vector")
4
5 RS = CorrelationMatrix(n_inputs)
6 RS[2, 3] = 0.75
7 R = NormalCopula.GetCorrelationFromSpearmanCorrelation(RS)
8 copula = NormalCopula(R)
9
10 myDist = ComposedDistribution(inpCollection,Copula(copula))
11 inp = RandomVector(Distribution(myDist))
12 model = Function(model_compute(names))
13 RandomGenerator.SetSeed(0)
14
15 ### polynomial chaos
16 pcmodel = Function(model)
17 pcmodel.setInputDescription(inputVariablesC)
18 pcmodel.setOutputDescription(outputVariablesC)
19
20 dim = inp.getDimension();
21 outP = CompositeRandomVector(pcmodel,inp)
22 polyColl = PolynomialFamilyCollection(dim)
23
24 # for lognormal distribution
25 lognormalFamily = StandardDistributionPolynomialFactory(AdaptiveStieltjesAlgorithm(dist_IOP))
26 polyColl[0] = OrthogonalUniVariatePolynomialFamily(lognormalFamily)
27 # for normal distribution
28 hermiteFamily = HermiteFactory()
29 for i in range(1,dim):
30 polyColl[i] = OrthogonalUniVariatePolynomialFamily(hermiteFamily)
31
32 multivariateBasis = OrthogonalProductPolynomialFactory(polyColl,EnumerateFunction(dim))
33
34 maximumConsideredTerms = 500
35 mostSignificant = 50
36 significanceFactor = 1.e-8
37 truncatureBasisStrategy = CleaningStrategy(OrthogonalBasis(multivariateBasis),
38 maximumConsideredTerms, mostSignificant, significanceFactor, True)
39 evaluationCoeffStrategy = LeastSquaresStrategy(LHSExperiment(sampleCreatePC))
40
41 polynomialChaosAlgorithm = FunctionalChaosAlgorithm(pcmodel, Distribution(myDist),
42 AdaptiveStrategy(truncatureBasisStrategy),ProjectionStrategy(evaluationCoeffStrategy))
43 polynomialChaosAlgorithm.run()
44 polynomialChaosResult = polynomialChaosAlgorithm.getResult()
45

```

```

46 coefficients = polynomialChaosResult.getCoefficients()
47 metaModel = polynomialChaosResult.getMetaModel()

```

We have implemented in Code 20.3 the construction of the PC metamodel; in particular the program:

- i. assemble the copula \tilde{K} (Eq. 18.8) following the information described in Sec. 1 \Rightarrow lines 1–8;
- ii. define the full model \Rightarrow lines 10 – 13;
- iii. define the PC expansion and the PC basis with the appropriate polynomial families following Tab. 18.1 \Rightarrow lines 16 – 32;
- iv. apply the truncation strategy and compute the PC expansion \Rightarrow lines 34 – 42;
- v. compute the PC coefficients \Rightarrow line 44;
- vi. define the metamodel based on PC expansion \Rightarrow line 45.

Code 20.4 – Test of the metamodel based on PC expansion

```

1 Xsample = myDist.getSample(size_testPC)
2
3 modelSample = model(Xsample)
4 metaModelSample = metaModel(Xsample)
5
6 np.save("Figures/sample_test.dat", Xsample)
7 np.save("Figures/sample_testModel.dat", modelSample)
8 np.save("Figures/sample_testMetaModel.dat", metaModelSample)
9
10 for output in range(n_outputs):
11     sampleMixed = Sample(size_testPC,2)
12     for i in range(size_testPC):
13         sampleMixed[i,0] = modelSample[i][output]
14         sampleMixed[i,1] = metaModelSample[i][output]
15
16     legend = str(size_testPC) + " realizations"
17     comparisonCloud = Cloud(sampleMixed, "blue", "fsquare", legend)
18     graphCloud = Graph('Polynomial chaos expansion '+str(outputVariablesC[output]),
19                       'model '+str(outputVariablesC[output]),
20                       'metamodel '+str(outputVariablesC[output]), True, 'topleft')
21     graphCloud.add(comparisonCloud)
22     graphCloud.draw('Figures/PCE_ModelsComparison_'+str(outputVariablesC[output]))

```

Before computing any Sobol index using the PC model we have to check that the metamodel is actually a good approximation of the full model. In Code 20.4 we

- i. create the input sample set to run the test \Rightarrow line 1;
- ii. generate the output for the full model \Rightarrow line 3;
- iii. generate the output for the full model \Rightarrow line 4;
- iv. save the input samples and the output results \Rightarrow lines 6 – 8;
- v. produce figures that compares the output of the two models \Rightarrow lines 10 – 20.

Code 20.5 – Compute the Sobol indices using the metamodel based on PC expansion

```

1 S1tilde_vec = FunctionalChaosRandomVector(polynomialChaosResult) # deprecated
2 S1tilde = FunctionalChaosSobolIndices(polynomialChaosResult)
3 s=[]
4 for output in range(n_outputs):
5     s.append([])
6     s[output].append(S1tilde_vec.getMean()[output])
7     print (" mean=",S1tilde_vec.getMean()[output])
8     s[output].append(np.sqrt(S1tilde_vec.getCovariance()[output,output]))
9     print (" variance=",np.sqrt(S1tilde_vec.getCovariance()[output,output]))
10    for i in range(dim):
11        s[output].append(S1tilde.getSobolIndex(i,output))
12        s[output].append(S1tilde.getSobolTotalIndex(i,output))
13        print (" ", inpCollection[i].getDescription()[0]," vs ", outputVariablesC[output],
14              "sobol index", S1tilde.getSobolIndex(i,output), "sobol total index",
15              S1tilde.getSobolTotalIndex(i,output))
16
17 sample_input = myDist.getSample(size)
18 sample_YPC = metaModel(sample_input)
19
20 np.save("Figures/sa.dat",s)

```

```

21 np.save("Figures/sample_ypc.dat", sample_YPC)
22 np.save("Figures/sample_input.dat", sample_input)
23
24 def column(matrix, i):
25     return [row[i] for row in matrix]
26
27 for output in range(n_outputs):
28     # Plot the sensitivity indices
29     w = 0.4
30     plt.figure(1+output*2)
31     #print("s=",s[output])
32     b1 = plt.bar((np.arange(dim)+1)-w, s[output][2:2*dim+1:2],
33                 width=w, color='#000999', label='Sobol Indices')
34     b2 = plt.bar((np.arange(dim)+1), s[output][3:2*dim+2:2],
35                 width=w, color='#66FFFF', label='Total Sobol Indices')
36     plt.legend(loc='upper right')
37     plt.xticks(list(np.arange(dim)+1), (r' IOP', r' RLTp', r' SP', r' DP'), size=8)
38     plt.yticks(list(np.linspace(0., 1., 5)))
39     plt.xlabel(output_names[output], fontsize=14)
40     plt.savefig("Figures/sa_"+output_names[output]+".pdf")
41     #
42     asampleYPC = np.array(column(sample_YPC, output)).flatten()
43     plt.figure(1+2*output+1)
44     plt.hist(asampleYPC, density=True, bins=int(np.floor(np.sqrt(size))))
45     plt.xlabel(output_names[output], fontsize=14)
46     plt.ylabel("Relative frequency", fontsize=14)
47     plt.savefig("Figures/pdf_"+output_names[output]+".pdf")

```

Finally Code 20.5 shows the computation of the Sobol indices using the algorithm explained in Sec. 2 and export the results *via* histograms for the Sobol indices and probability density functions of the output variables.

3 Results

In all the experiments, we show the results for the following output:

- i) *CRA*: the simulated maximum blood flow within the CRA (Fig. 20.2);
- ii) *CRV*: the simulated maximum blood flow within the CRV (Fig. 20.3);
- iii) *LC*: the simulated maximum blood flow within the lamina cribrosa (Fig. 20.4).

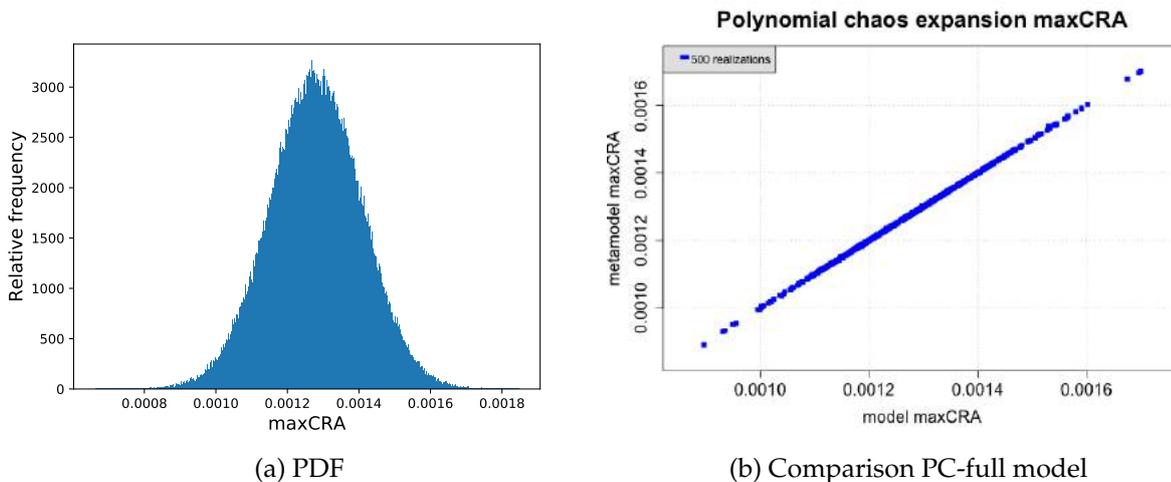


Figure 20.2 – *Central Retinal Artery* probability density function (PDF) and comparison result between the PC expansion model and the full model.

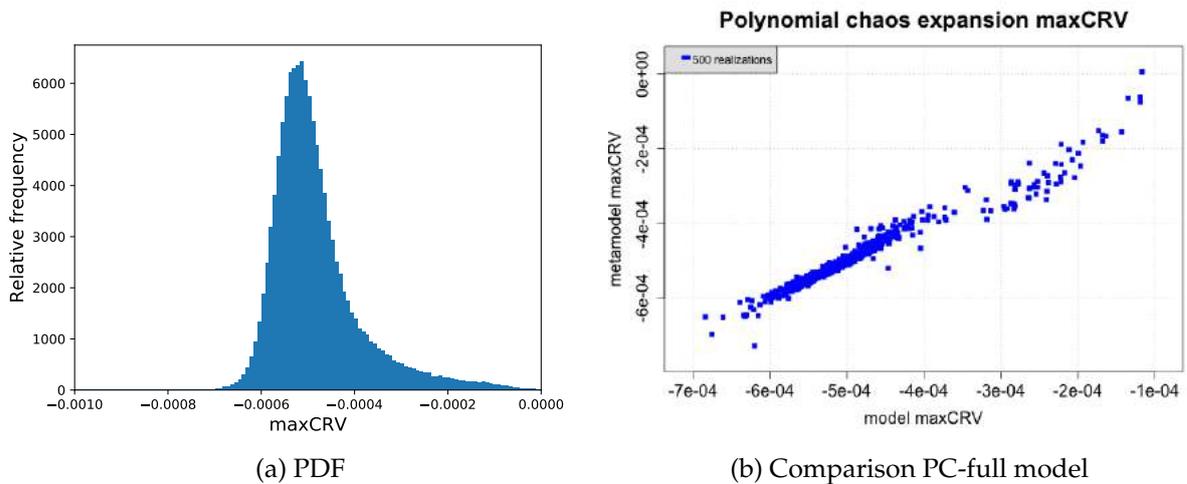


Figure 20.3 – *Central Retinal Vein* probability density function (PDF) and comparison result between the PC expansion model and the full model.

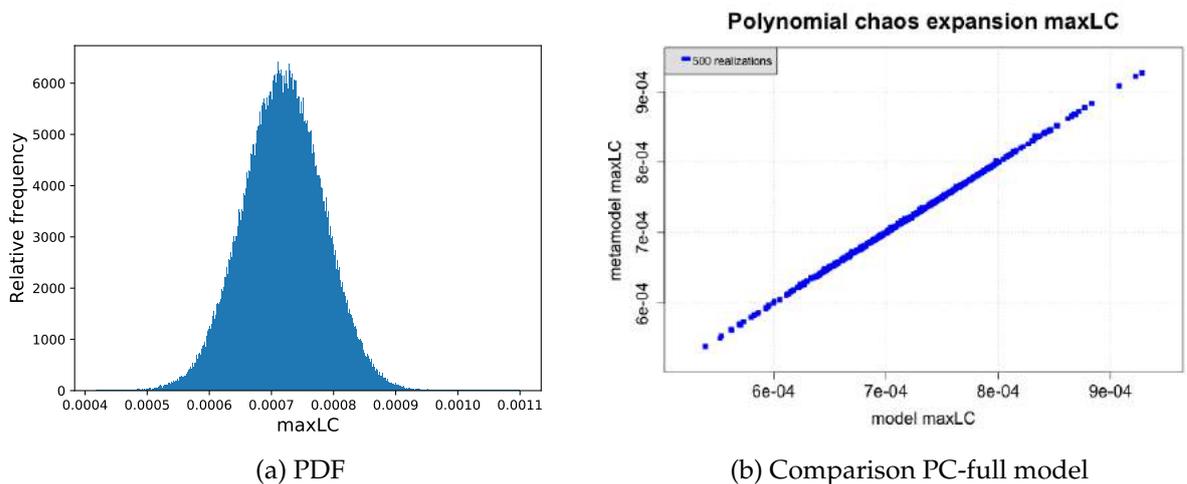


Figure 20.4 – *Lamina cribrosa* probability density function (PDF) and comparison result between the PC expansion model and the full model.

For what concerns the blood flow within the CRA, we have a Gaussian profile for its probability density function (Fig. 20.2a), with a median of $0.001279 \text{ cm}^3/\text{s}$, which agrees with the experimental value reported by Dorner et al. [62] ($0.00131463636 \text{ cm}^3/\text{s}$). Fig. 20.2b shows the results of the test between the full model and the polynomial chaos expansion. With 500 realizations perfectly aligned on the diagonal, we have ensured that the reduced model captures the behavior of the CRA blood flow as for the original full model. Thus, the Sobol indices analysis presented in Fig. 20.5a is accurate and points out a high dependency of the CRA on the systemic systolic blood pressure, a small reliance on the IOP and a negligible influence by RLTP and DP. These results are reasonable if we remind that the arteries are stiff tubes, therefore external pressures should not impact the blood flow within these vessels (IOP and RLTP), whereas also intuitively the systolic pressure at inflow is the dominant parameter for the variability of the CRA maximum blood flow.

The probability density function of CRV maximum blood flow (Fig. 20.3a) looks like a log-normal distribution, which is acceptable from a physiological viewpoint [50]. The comparison

between the full model and the PC expansion illustrated by Fig. 20.3b is not as good as the previous one, however it presents a satisfying pattern diagonal-like. In contrast with the CRA, the CRV is a venous vessel, which can be influenced both by the blood profile at inflow and external pressures; indeed the Sobol index analysis in Fig. 20.5b reveals that the two higher indices are the systemic systolic pressure and the IOP. Both indices shows also significant differences between the first-order Sobol index (blue bar) and the total-order index (turquoise bar), which means that there are higher-order interactions between these parameters. This fact is not surprising if we remind our modeling choice to represent the venous part with Starling resistors (Eq. (3.8)).

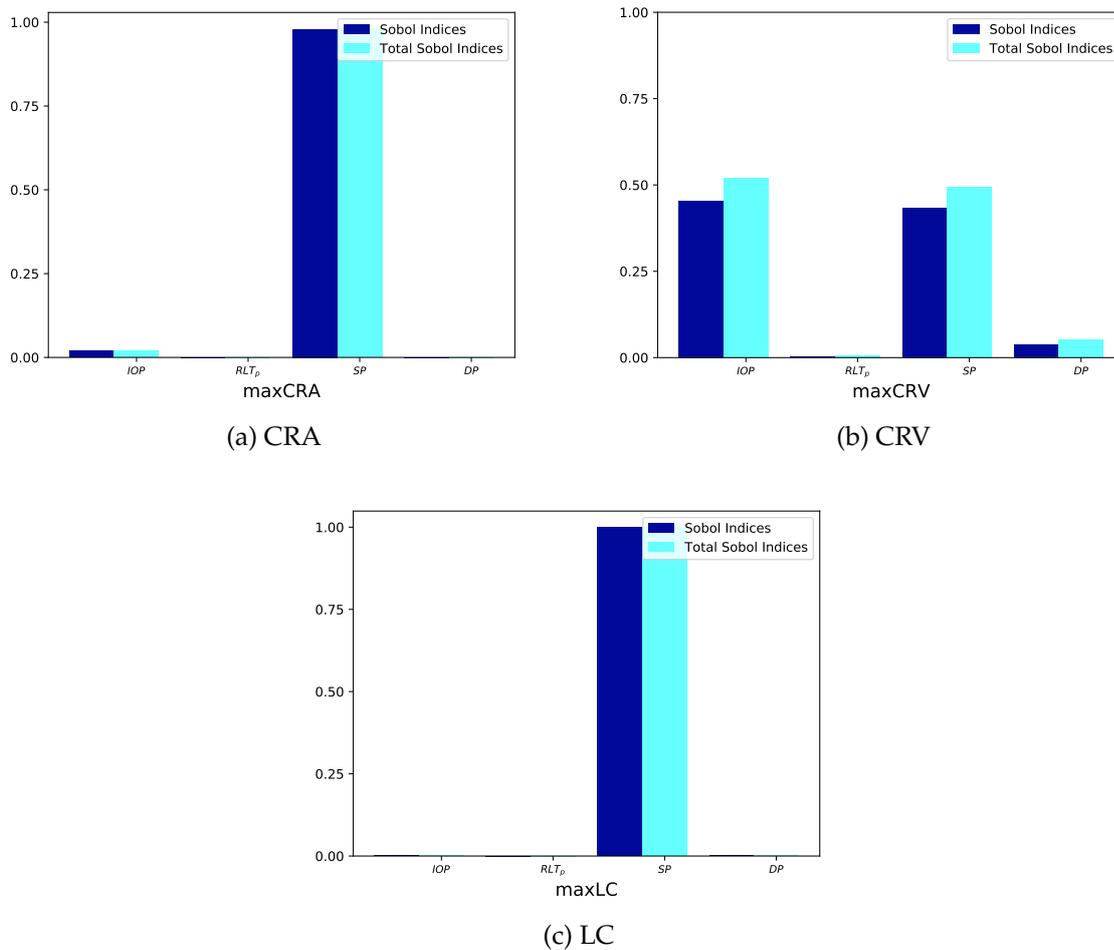


Figure 20.5 – Computed Sobol indices of the output variables.

For the blood flow within the lamina cribrosa, the results are similar to the CRA ones: the LC probability density function is a Gaussian distribution (Fig. 20.4a), very good agreement between the PC expansion and the full model (Fig. 20.4b), and the Sobol index analysis shows a high dependency on SP. This last outcome may be not so intuitive as for the other two output, indeed we know that the pressure gradient across the LC ($IOP - RLT_p$) may influence the hemodynamics [210]. In our opinion, this particular behavior is due to lack of pure biomechanics in the simple initial model we have used for this analysis (model described in Ch. 3), whereas, for instance, already with a model like the described in Ch. 5 we should notice the impact of the IOP and the RLT_p .

4 Conclusions

The proposed sensitivity analysis using Sobol indices has shown the potentiality of our model to capture the realistic physiological behavior of the ocular vasculature. Rigorously, we have applied the theory of uncertainties quantification in order to build a reduced model using the PC expansion and to compute the Sobol indices of some significant output.

This mathematical and computational tool is now set up to perform stochastic analysis on more complex models following the directions dictated by ophthalmology research.

Part VII

Future perspectives

In this part we present three projects that are not directly related to the title of my PhD, however, in perspectives, they may be useful to extend the current work:

1. a review of the mathematical models that describes the cerebrospinal fluid (CSF), presented in Ch. 21
2. a study of the ballistocardiogram (BCG) and the mathematical model we have developed in order to describe the human cardiovascular system, illustrated in Ch. 22;
3. the characterization of a transistor in inversion condition, detailed in Ch. 23;
4. a traction test study to understand the effect of IBC for linear elastic equations, proposed in Ch. 24.

The CSF review is interesting for the OMVS since at the moment the RLTP is an input value that has to be chosen to feed the model; however a mathematical model that describes the dynamics of the CSF would provide directly to our model the pressure value behind the lamina cribrosa.

For the BCG project we developed a lumped-parameter model that describes the human body circulation with a particular attention on the functioning of the heart. This circuit can be interfaced with our *Level 0* of the OMVS and, thus, integrated in the overall mathematical model to obtain, although approximated, a closed-loop blood circulation in which we may simulate the effects of some systemic diseases, such as diabetes, heart failure or hypertension, on the eye.

In the transistor application we have employed the HDG method described in Ch. 8 for an industrial application, in particular in the context of semiconductor devices.

For the last study, we use the HDG method for linear elasticity to compare the simulation results of a virtual uniaxial test when imposing Neumann or integral BCs.

Mathematical modeling of the cerebrospinal fluid flow

Contents

1	Reduced models for CSF dynamics	222
2	Distributed models for CSF dynamics	226

The aim of this chapter is to provide a brief overview of mathematical models that describe the CSF and its possible interactions with other biofluids or neighboring tissues.

The CSF is a clear, colorless liquid that fills and surrounds the brain and the spinal cord and provides a mechanical barrier against shock. Even if it is not clear the exact method of formation, the CSF is formed mainly from arterial blood by the choroid plexuses of the lateral and fourth ventricles by a combined process of diffusion, pinocytosis and active transfer with a production rate of 0.2 – 0.7 ml per minute or 600 – 700 ml per day. The total volume of CSF in the adult ranges from 140 to 270 ml and its circulation is aided by the motion of the cilia of ependymal cells and by the pulsations of the choroid plexus. Thus, CSF is absorbed across the arachnoid villi into the venous circulation and a significant amount probably also drains into lymphatic vessels around the cranial cavity and spinal canal with an absorption rate that correlates with CSF pressure.

It supports the brain and provides lubrication between surrounding bones and the brain and spinal cord, in particular when an individual suffers a head injury, the fluid acts as a cushion, dulling the force by distributing its impact. CSF also plays an important role in the homeostasis and metabolism of the central nervous system, maintaining a constant pressure within the cranium. For further details on the CSF physiology please refer to [92, Ch. *Anatomy and Physiology of the Cerebrospinal Fluid.*]

The analysis of this brain fluid is crucial to diagnose a certain number of diseases. The currently method more adopted is the lumbar puncture, which consists in obtaining a fluid sample by inserting a needle into the lumbar region of the lower back below the termination

of the spinal cord. This method is quite invasive and it may be affected by the position of the patient (supine), however novel non-invasive methods are coming into place in order to measure the intracranial pressure (ICP) [223, 197, 196, 159].

For a detailed discussion of this topic see [92, Ch. *Mathematical modeling of the cerebrospinal fluid flow and its interactions.*].

We focused in the description of mathematical models based on the fundamental laws of physics, *e.g.* mass and momentum conservation. These models can be multiscale both in space and in time and multiphysics (i.e fluid-structure coupling).

Our interest is on models of intracranial dynamics, since this is the most meaningful part in view of modeling ocular fluid dynamics, without neglecting coupled cranial-spinal models. We have only briefly reviewed the “stand-alone” spinal interactions.

In addition to that, we reviewed macroscopic mathematical models available to study CSF flow, therefore chemistry and both micro- and mesoscale models will not be considered. Whenever possible, we highlight the contributions to clinical applications for all the models reviewed, in particular in the field of neurological pathologies (Hydrocephalus, Chiari Malformation etc.).

CSF dynamics and its complex coupling with the other cerebral fluids and surrounding tissues was recently reviewed in:

- V. Kurtcuoglu [132].
- Linninger et al [144].

Our aim is to specify a complementary view to these works by providing a review of selected works from the abundant literature on CSF modeling based on the possible connections between these brain models with existing models in ocular fluid dynamics. The chapter is organized as follows:

- Sec. 1 is devoted to the description of the reduced modeling approach and to the review of different contributions in this direction;
- Sec. 2 focuses on distributed models and on reviewing several contributions in the literature following this approach.

1 Reduced models for CSF dynamics

This section reviews several *reduced* models for CSF flow, possibly coupled with cerebral hemodynamics and brain tissue dynamics. We also present two very recent contributions studying the complex interplay between cerebral and ocular biofluids.

Reduced models have the advantage of being a useful tool to provide insights on the interaction between different constituents and/or compartments of the system under investigation. In addition, they are able to capture the main physical phenomena governing the system at a lower computational cost than the full solution of a three-dimensional problem describing the CSF flow by means of a partial differential equation (PDE) model.

Various modeling reduction techniques have been proposed in the literature, see for instance [72, Ch. 10] and references cited therein. *One-dimensional models* (1D) take advantage of the basically cylindrical morphology of some biological conduits. The resulting mathematical description is based on 1D hyperbolic PDEs, which can capture wave-propagation phenomena in the biofluids, with a reasonable computational cost. This is not the approach we will follow in the sequel, since we rather focused on an even more simplified description of the flow.

Reduced *zero-dimensional models* (0D), also called *lumped parameter models*, provide a circuit-based representation of the fluid dynamics in each compartment, based on the analogy between electric and hydraulic networks. By writing *Kirchhoff laws* for the nodes (conservation of current/flow rate) and for closed loops (conservation of the voltage/pressure), the resulting mathematical model is a system of differential algebraic equations (DAE), potentially nonlinear. These models are used in different ways depending on the specific modeling needs.

In a first approach, many studies implemented them as boundary conditions for three-dimensional simulations in regions of particular interest. Alternatively, stand-alone 0D reduced models have been used to understand the main dynamics of the system thanks to their simplified mathematical structure and to compute average values of the unknowns. They are also particularly well suited for sensitivity analysis studies with respect to changes in the input data that can be used to discriminate the contribution of different factors on flow quantities. In addition, these robust mathematical models have the advantage that their complexity can be dynamically increased and implemented as the understanding of the underlying phenomena evolves.

The main modeling ingredients involved in the mathematical description of 0D reduced models for fluid flows are:

- *Hagen-Poiseuille's law*, according to which the *pressure-driven flux* Q_{ij} between the generic compartments i and j is governed by the hydraulic analogue of Ohm's law

$$Q_{ij} = \frac{P_i - P_j}{R_{ij}} \quad (21.1)$$

where R_{ij} denotes the hydraulic resistance between compartment i and compartment j ;

- *Starling-Landis's law*, describing *filtration-driven flows* in general terms as follows: the flux Q_{ij} due to filtration from compartment i to compartment j is modeled by

$$Q_{ij} = K_{ij}[(P_i - P_j) - \sigma_{ij}(\pi_i - \pi_j)], \quad (21.2)$$

where π_i is the osmotic pressure in i , P_i is the mean hydraulic pressure in the compartment i , K_{ij} is the filtration coefficient from i to j and σ_{ij} is the corresponding reflection coefficient.

- *Darcy's law*, characterizing *diffusive flows* in porous media according to the following expression:

$$Q_{ij} = \frac{kA}{\mu L} (P_i - P_j), \quad (21.3)$$

where k is the permeability of the medium, A is the the cross-sectional area to flow, μ is the viscosity of the fluid, and L the characteristic length over which the pressure drop $P_i - P_j$ takes place.

For a more in-depth discussion on the above equations, their interpretation and the parameters involved in their description, see [72, Ch. 10] and [92, Ch. 2].

The mathematical translation of the *clinical concepts* of CSF production, circulation, absorption and storage started in the second part of the 20th century, and early works such as [57], [95] and especially [153], [241] are still used in contemporary clinical neuroscience. These contributions put the basis of the theoretical study of three *basic clinical maneuvers* – bolus CSF withdrawal, addition and constant rate infusion – and allowed to characterize CSF circulation disorders using parameters from the models, such as resistance to CSF outflow, elasticity and pressure-volume index (PVI). For example, the Servo-Controlled Constant Pressure Infusion Test [65] based on [153] is used for the assessment of hydrocephalus and progressive dementia. PVI is a parameter utilized to describe CSF compensation in hydrocephalic subjects [127] or traumatic brain injury [220]. Subsequently, the mathematical models describing CSF were enriched to incorporate autoregulation, interactions with blood flow, tissues or other biofluids and many of these ideas are still actively investigated in current research. In Tabs. 21.1 and 21.2 we give an overview of some significant contributions in the literature.

REFERENCE	PURPOSE	CLINICAL RELEVANCE
CEREBRAL CSF HYDRODYNAMICS		
Marmarou1978 [154]	ICP dynamics in terms of nonlinear intracranial compliance [153], dural sinus pressure, CSF formation, and resistance to CSF absorption.	Assessment of the factors leading to elevation of ICP and identification of clinical descriptors useful as therapeutic and prognostic guides.
CEREBROSPINAL HEMO-HYDRODYNAMICS		
Ambarki2007 [4]	CSF and blood flows interactions with volume changes during the cardiac cycle.	Study of CSF and blood flow oscillation during cardiac cycle at different sites and their kinetic energy impact at their interface boundary.
Gehlen2017 [79]	CSF volume and compliance for different hydrostatic pressure gradients between the CSF and venous system.	Prediction of craniospinal compliance distribution shift and CSF pulsation changes related to postural changes.

Table 21.1 – Reduced models: overview.

REFERENCE	PURPOSE	CLINICAL RELEVANCE
CEREBRAL HEMO-HYDRODYNAMICS		
Ursino1997 [241]	ICP dynamics interacting with cerebral blood volume changes and autoregulation.	Study of ICP in pathological conditions (brain injury, subarachnoid hemorrhage, hydrocephalus, brain tumor) on the basis of data from routine clinical measurements, <i>e.g.</i> PVI tests.
CEREBRAL HEMO-HYDRODYNAMICS AND BRAIN DYNAMICS		
Lakin2003 [134]	ICP, volume and flow dynamics in a whole body physiology (blood, CSF, interstitial fluid, lymph, cerebrovascular autoregulation and regulation by the sympathetic nervous system).	Study of the effects of CSF infusion, change in body position and pathological conditions (cardiac arrest, hemorrhagic shock).
Linninger2009 [145]	ICP dynamics between cerebral vasculature, biphasic brain parenchyma, ventricular system (extended from [143]), and compliant spine.	Prediction of intracranial pressure gradients, blood and CSF flow and displacements in normal and pathological conditions (hydrocephalus).
Buishas2014 [24]	CSF dynamics as a function of both hydrostatic and osmotic pressure gradients between cerebral vasculature, ECS, perivascular space and CSF. Both the classical and the microvessel hypothesis of CSF production and reabsorption are included.	Assesment of the effects of the osmolarity of ECS, blood, and CSF elucidating the mechanism of water exchange occurring in the brain (osmotic imbalances, hydrocephalus, edema).
CEREBRAL AND EYE HEMO-HYDRODYNAMICS		
Nelson2017 [166]	Blood and aqueous humor dynamics, lamina cribrosa biomechanics and IOP-dependent ocular compliance affecting volume/pressure in the eye.	Prediction of the impact on IOP due to increased ocular blood pressures and ICP, associated with short term effects induced by gravitational variations.
Salerni2018 [213]	Hemodynamics in the eye (retina, choroid, ciliary body, lamina cribrosa) and brain interacting with the dynamics of aqueous humor and CSF; lamina cribrosa biomechanics influenced by tissue pressure in the optic nerve head due to CSF within the SAS.	Assessment of the role of various factors and the mechanism of their interactions implicated in the loss of visual function in various condition, <i>e.g.</i> ocular pathologies, long head-down tilt experiment, microgravity environment.

Table 21.2 – Reduced models: overview.

2 Distributed models for CSF dynamics

In this section we discuss the main challenges in the mathematical and computational modeling of CSF flow towards realistic geometries and to review several contributions within this framework. We focus on the so-called *distributed* (or macro-scale) *models* that are suitable when aiming for a three-dimensional representation of the flow. The description of the underlying mechanisms stems from the basic principles of fluid dynamics and it is translated into systems of PDEs, supplied with suitable initial and boundary conditions of particular importance. Specifically, CSF is often considered as an incompressible Newtonian fluid, therefore the *Navier-Stokes equations* are employed (see for instance [72, Ch. 2, Ch. 3]). In addition, the fluid dynamics description can be enriched to take into account the combined effects of flow and different structures from a *multiphysics* perspective including *poroelasticity* [72, Ch. 7] and *fluid-structure interaction* [72, Ch. 3, Ch. 9].

The issue of boundary conditions when modeling biological fluids is of major importance and matter of intense research, from the mathematical and computational viewpoint. When performing large scale three-dimensional simulations, the domain has to be reduced to a region of interest because of the prohibitive computational costs. Therefore, its boundary is composed of two parts: a *physical* boundary - corresponding to the geometrical description of the compartments - and an *artificial* boundary - at the level where the domain is truncated. On the one hand, two possibilities can be considered for the physical boundaries: either assume them rigid, in which case the no-slip Dirichlet condition is considered, or incorporate their ability to deform, and thus include a much more complex fluid-structure behavior in the model. On the other hand, at the artificial boundaries, the formulation should be able to take into account the rest of the closed circuit representing the CSF circulation, which is still a very challenging issue, as discussed for instance in the recent review [195]. Moreover, in order to achieve physiological simulations, data used to impose the boundary conditions should be taken from clinical measurements.

In this perspective, the geometrical representation of the CSF space is very complex and CSF flow quantification requires special care. Recent progress in segmentation of vascular volumes from magnetic resonance imaging (MRI) data, design of a vascular models and construction of computational meshes of the vascular structures, as well as PC-MRI techniques for blood flow measurements are translated into use for CSF flows. However, the CSF compartments have their own features, that are contributing to the difficulty of deriving accurate information, among which: (i) complex geometric shapes, in contrast to the (locally) near-cylindrical form of blood vessels; (ii) smaller magnitudes in the velocity field compared to blood flow; (iii) difficulties in finding planes perpendicular to the flow direction; (iv) presence of reversal flow, *etc.*

The understanding of CSF behavior at *fine scales* is of major importance, since abnormal flow dynamics or pressures imbalances in different compartments may lead to several pathologies. A recent compilation of studies on abnormal CSF dynamics in central nervous system (CNS) diseases can be found in [144], with a focus on some of the most common ones: Hydrocephalus, Chiari Malformation Type 1, Syringomyelia, Pseudotumor Cerebri, Idiopathic Intracranial Hypertension and Benign Intracranial Hypertension. The pathophysiology and the etiology of these syndromes are still poorly understood and trigger a lot of open research questions. In addition, the clinical manifestations, diagnosis and management strategies are currently a matter of active discussion. In perspective, a better understanding of both normal and pathological

CSF flow thanks to mathematical and computational models encompassing the whole 3D dynamics is very appealing, since it may be able to provide not only a qualitative assessment of the dynamics, but also a more quantitative knowledge about CSF velocities, flow rates and pressures.

The long-term goal is to devise a complete *in silico* model of CNS dynamics, accounting for the full coupling between CSF, blood and brain tissues and incorporating various physical properties and settings (*e.g.* variations of body position, patient-specific parameters and geometries, *etc.*). Furthermore, to the best of our knowledge, a full 3D mathematical and computational modeling of the complex connections between biofluids in the coupled eye-brain system is still not available and only a few recent reduced models tackled this challenging issue. Such a very complex formulation would require large-scale availability of data, sophisticated mathematical and numerical methods and high computational costs. Several recent contributions that we will review in the sequel made significant steps in this direction and provided new insights to the understanding and better management of CSF flow conditions: Chiari Malformation (Bunck et al. [25], Støverud et al. [231], Jain et al. [118]), hydrocephalus and cerebral oedema (Vardakis et al. [243]), drug delivery in Leptomeningeal dissemination, epilepsy, Parkinson (Howden et al. [108]) to give just a few examples.

In Tabs. 21.3, 21.4 and 21.5 we give an overview of some significant contributions in the literature.

REFERENCE	PURPOSE	CLINICAL RELEVANCE
Jacobson1996 [117]	Computation of CSF flow in idealized cylindrical domains.	Estimation of the pressure drop required to achieve normal physiological CSF flow rate within the aqueduct of Sylvius.
Kurtcuoglu2007 [133]	Investigation of subject-specific CSF flow characteristics by combining MRI scans and CFD simulations.	Quantitative prediction of the subject-specific flow field in the third ventricle and the aqueduct of Sylvius.
Gupta2010 [96]	Prediction of subject-specific CSF flow in the cranial SAS using <i>in vivo</i> data.	Quantitative and qualitative description of cranial CSF transient flow, potentially useful to compare healthy with pathological CSF dynamics.

Table 21.3 – Spatially distributed models: overview.

REFERENCE	PURPOSE	CLINICAL RELEVANCE
Howden2011 [108]	Prediction of the CSF flow in the CNS, the SAS, and in the ventricular system, with highlights in the aqueduct of Sylvius, the exit of Magendie, and entrance to the Spinal SAS.	Investigation of the role of the cardiac cycle, of the pulsatility and of the porous arachnoid trabeculae within the SAS on the fluid motion of the CSF by a global 3D model of the CNS.
Sweetman2011 [235]	Prediction of 3D CSF velocity and pressure fields in the CNS, comparison with <i>in vivo</i> measurements from a healthy human subject.	Quantification of normal intracranial dynamics, potentially significant to analyze diseased intracranial dynamics.
Hadzri2011 [97]	Investigation of 3D CSF flow dynamics in a stenosed aqueduct.	Inclusion of a stenosis inducing significantly increased pressure drop; possible clinical application to monitor hydrocephalus.
Bunck2012 [25]	Statistical analysis on 3D data for a population base study: 20 Chiari malformation patients and 10 healthy volunteers.	Comparison between healthy subjects and Chiari I patients, showing qualitative and quantitative alterations of CSF flow, in particular for patients with associated syringomyelia.
Hsu2012 [109]	Construction of a drug delivery model through the SAS and spinal cord tissue.	Identification of patient-specific key variables (frequency and magnitude of CSF pulsations) on drug distribution at the spine level.
Rutkowska2012 [206]	Visualization of CSF flow simulation in patient-specific models of the SAS.	Comparison of CSF dynamics between healthy subjects and Chiari I patients both with CFD simulations and PC-MRI results.
Tangen2015 [236]	Quantification of the contribution of micro-anatomical aspects on CSF flow patterns and flow resistance within the entire CNS.	Investigation of complex CSF dynamics in a subject-specific model of the entire CNS, including micro-anatomical aspects and their impact on drug distribution.
Vardakis2016 [243]	Development of a novel spatio-temporal model of fluid regulation and cerebral tissue displacement in the framework of Multiple-Network Poroelastic Theory.	Investigation of interstitial oedema formation and its alleviation, for a better understanding of diseases like hydrocephalus or cerebral oedema.

Table 21.4 – Spatially distributed models: overview.

REFERENCE	PURPOSE	CLINICAL RELEVANCE
Støverud2016 [231]	Construction of a subject-specific model to compare CSF flow variations in the cervical SAS and in the SAS of the posterior cranial fossa.	Comparison of CSF flow differences between Chiari I patients and healthy individuals.
Jain2016 [118]	Quantification of CSF hydrodynamics down to very fine scales to investigate the possible occurrence of transitional and turbulent CSF flows.	Comparison between CSF flow in healthy and Chiari Malformation I patients, with a special focus on the cranio-vertebral junction.
Garnotel2017 [78]	Construction of a simplified model to understand the dynamics of the CSF, its impact on ICP and CBF.	Description of CSF distribution in cerebral ventricles, cerebral SAS and spinal SAS, accounting for the confined environment of the skull and the synchronization with the cardiac cycle.

Table 21.5 – Spatially distributed models: overview.

 Interpretation of cardiovascular function via the ballistocardiogram

Contents

1	Background and related work	233
2	Mathematical Formulation of Closed-Loop Model	234
3	Results	241
4	Theoretical and experimental ballistocardiograms	244
5	Conclusions	246

In this chapter we discuss a project in the context of ballistocardiography, which content has been published in [94]. Ballistocardiography captures the signal generated by the repetitive motion of the human body due to sudden ejection of blood into the great vessels with each heart beat [229]. The signal is called the ballistocardiogram (BCG). Extensive research work by Starr and Noordergraaf showed that the effect of main heart malfunctions, such as congestive heart failure and valvular disease, would alter the BCG signal [229, 182, 230], thereby yielding a great potential for passive, non-contact monitoring of the cardiovascular status.

The original measurement device used by Starr and others was a lightweight bed suspended by long cables. The blood flow of a subject lying on the suspended bed resulted in the bed swinging; the capture of the swing was the BCG signal. This measurement device was impractical for standardized BCG measurements, especially compared to the electrocardiogram (ECG), which could be taken on virtually any platform using electric leads placed on the body in a standard configuration. The standardization of the ECG measurement has allowed clinical interpretation of the ECG waveform, such that it can be used to diagnose abnormalities in cardiac function.

Recently, there has been a resurgence of BCG research, as new sensing devices (*e.g.* in the form of bed sensors) allow easier, noninvasive capture of the BCG signal. In addition, the BCG offers advantages over the standardized ECG measurement, in that direct body contact is not

required and the signal reflects the status of the larger cardiovascular system. These advantages provide intriguing possibilities of continuous passive monitoring of the cardiovascular system without requiring the patient to do anything.

There have been a variety of bed, chair, and other sensors proposed to capture the BCG; several are now available commercially [113, 221, 112, 82, 201, 2, 177, 37, 123, 205, 110, 257, 261, 105]. Much of this work has focused on monitoring heart rate along with respiration rate from the accompanying respiration signal, and other parameters for tracking sleep quality. Recent work has also investigated the BCG waveform morphology for the purpose of tracking changes in cardiovascular health [232, 182, 119]. This offers a special relevance and significant potential in monitoring older adults as they age. Identifying very early signs of cardiovascular health changes provides an opportunity for very early treatments before health problems escalate; very early treatment offers better health outcomes and the potential to avoid hospitalizations [198, 199].

One challenge in using the BCG waveform to track cardiovascular health changes is the lack of a standardized measurement device and protocol and, thus, the lack of uniform clinical interpretation of the BCG signal across the various sensing devices. Our project is a first step in addressing this challenge by building a theoretical BCG signal based on a novel closed-loop mathematical model of the cardiovascular system and comparing the simulated output with the BCG signal measured by accelerometry on a suspended bed. Parameters of the model are calibrated on known physiological conditions for the average human body.

In the following section we will present our contribution in the mathematical model, whereas the description of the sensing bed used for the experiment can be found in [94]; indeed the validation of the proposed mathematical has been realized thank to a replica of Starr's suspended bed built within the Center for Eldercare and Rehabilitation Technology at the University of Missouri (see Fig. 22.1).



Figure 22.1 – Replica of the original Starr's suspended bed built within the Center for Eldercare and Rehabilitation Technology at the University of Missouri.

1 Background and related work

Staar and Noordergraaf provided the theoretical foundations to interpret BCG signals by expressing the displacement of the center of mass of the human body as a function of the blood volumes occupying different vascular compartments at a given time during the cardiac cycle [229]. Specifically, the coordinate Y of the center of mass of the body along the head-to-toe direction at any given time t can be written as

$$Y(t) = \frac{\rho_b}{M} \sum_{i=1}^N V_i(t) y_i + c \quad (22.1)$$

where ρ_b is the blood density, M is the body mass, N is the total number of vascular compartments considered in the model and c is a constant term representing the body frame. Each vascular compartment i , with $i = 1, \dots, N$, is assumed to be located at the fixed coordinate y_i and to be filled with the blood volume $V_i(t)$ at time t . Since the term c in Eq. (22.1) is constant for a given person, the BCG signal associated with the center of mass displacement in the head-to-toe direction is defined as

$$\text{BCG}_{disp}(t) := \frac{\rho_b}{M} \sum_{i=1}^N V_i(t) y_i. \quad (22.2)$$

The BCG signals associated with velocity and acceleration of the center of mass can be obtained via time-differentiation as

$$\text{BCG}_{vel}(t) := \frac{\rho_b}{M} \sum_{i=1}^N \frac{dV_i(t)}{dt} y_i, \quad (22.3)$$

$$\text{BCG}_{acc}(t) := \frac{\rho_b}{M} \sum_{i=1}^N \frac{d^2V_i(t)}{dt^2} y_i, \quad (22.4)$$

respectively. In our body, the waveforms $V_i(t)$ result from the complex interplay between the blood volume ejected from the heart, the resistance to flow that blood experiences across the cardiovascular system and the pressure distribution within it. Staar and Noordergraaf characterized the volume waveforms $V_i(t)$ by means of experimental measurements at each location y_i [229].

We have calculated the waveforms $V_i(t)$ by means of a mathematical model based on the physical principles governing vascular physiology, thereby paving the way to the use of quantitative methods to interpret BCG signals and identify cardiovascular abnormalities in a given patient.

The first computer-aided approach for quantitative interpretation of BCG signals was proposed by Noordergraaf et al in [173], where the electric analogy to fluid flow was leveraged to describe the motion of blood through the arterial system during the cardiac cycle and calculate the resulting BCG signal. Since then, only a few studies have been directed to the theoretical interpretation of BCG signals.

In [250], Wiard et al utilized a three-dimensional finite element model for blood flow in the thoracic aorta to show that the traction at the vessel wall appears of similar magnitude to recorded BCG forces.

In [126], Kim et al proposed a simplified model based on the equilibrium forces within the

aorta to show that blood pressure gradients in the ascending and descending aorta are major contributors to the BCG signal.

Despite their different approaches to blood flow modeling, the aforementioned studies share the common feature of focusing only on the arterial side of the cardiovascular system, thereby leading to *open-loop* models of the circulation. In reality, though, our blood circulates within a *closed-loop* system and, as a consequence, hemodynamic changes observed at the level of the major arteries might be a result of changes occurring elsewhere within the closed-loop system. For example, left ventricular heart failure leads to an increase in fluid pressure that is transferred back to the lungs, ultimately damaging the right side of the heart and causing right heart failure [19].

The novelty in this work is, for the first time, a quantitative interpretation of the BCG signal by means of a *closed-loop* mathematical model for the cardiovascular system. Several modeling approaches have been proposed in cardiovascular research, see *e.g.* [124, 227, 258]. In particular, some work have highlighted the non-negligible effect of the feedback wave within a closed-loop circuit, see *e.g.* [140, 176, 16, 107]. We have leveraged the knowledge available in the modeling literature to develop a novel closed-loop model for the cardiovascular system that includes sufficient elements to reproduce theoretically the BCG signal.

2 Mathematical Formulation of Closed-Loop Model

The closed-loop model depicted in Fig. 22.2 can be represented mathematically by a system of ODEs of the form

$$\mathcal{M}(Y(t)) \frac{dY(t)}{dt} = \mathcal{A}(Y(t))Y(t) + b(Y(t)) \quad t \in [0, T] \quad (22.5)$$

where Y is the m -dimensional column vector of unknowns, \mathcal{M} and \mathcal{A} are $m \times m$ tensors and b is the m -dimensional column vector of given forcing terms. We anticipate that \mathcal{M} , \mathcal{A} and b nonlinearly depend on the vector of state variables Y . The specific expressions for Y , \mathcal{M} , \mathcal{A} and b follow from the constitutive equations characterizing the circuit elements and the Kirchoff laws of currents and voltages. The model leverages the electric analogy to fluid flow, where resistors, capacitors and inductors represent hydraulic resistance, compliance and inertial effects, respectively. The model comprises 4 main interconnected compartments:

- (i) the heart, where the valves are represented by ideal switches and the ventricular pumps are modeled by means of voltage sources;
- (ii) the systemic circulation, where the arterial segments from the ascending aorta to the iliac arteries are modeled in great detail because of their relevance to the BCG signal;
- (iii) the pulmonary circulation, which starts with the pulmonary artery at the right ventricle and converges into the left atrium;
- (iv) the cerebral circulation, which branches from the aortic arch and converges into the systemic venous system.

The types of electrical elements included in the closed-loop model are summarized in Fig. 22.3. The constitutive laws characterizing each element are detailed below. For the sake of clarity we reintroduce the elements that have been already presented in Ch. 3.

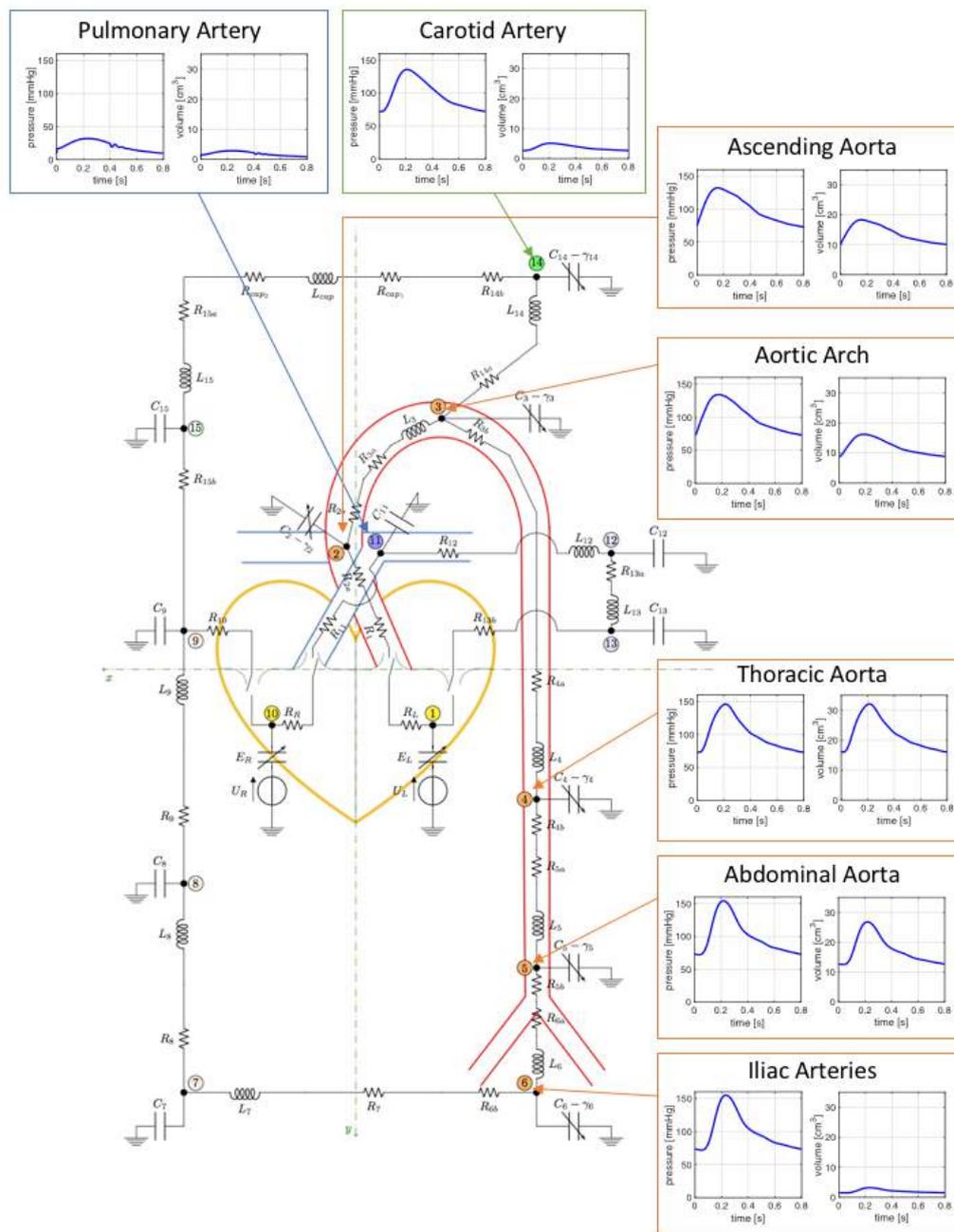


Figure 22.2 – Schematic representation of the closed-loop mathematical model describing the flow of blood through the circulatory system. The circuit nodes have been marked by thick black dots and numeric labels from 1 to 14 enclosed in circles of different colors (yellow, orange, blue, green) to distinguish between the 4 circulatory compartments included in the model (heart, systemic circulation, pulmonary circulation, cerebral circulation, respectively). Numerical labels in fuller color tones identify the circuit nodes utilized for the computation of the BCG signal. Arterial pressure and volume waveforms simulated via the closed-loop model are also included.

- *Linear resistor* (see Fig. 22.3(a)): the hydraulic analog to Ohm’s law states that the pressure difference $P_0 - P_1$ across the resistor is proportional to the volumetric flow rate Q , with the hydraulic resistance R as proportionality constant, namely

$$P_0 - P_1 = R Q . \tag{22.6a}$$

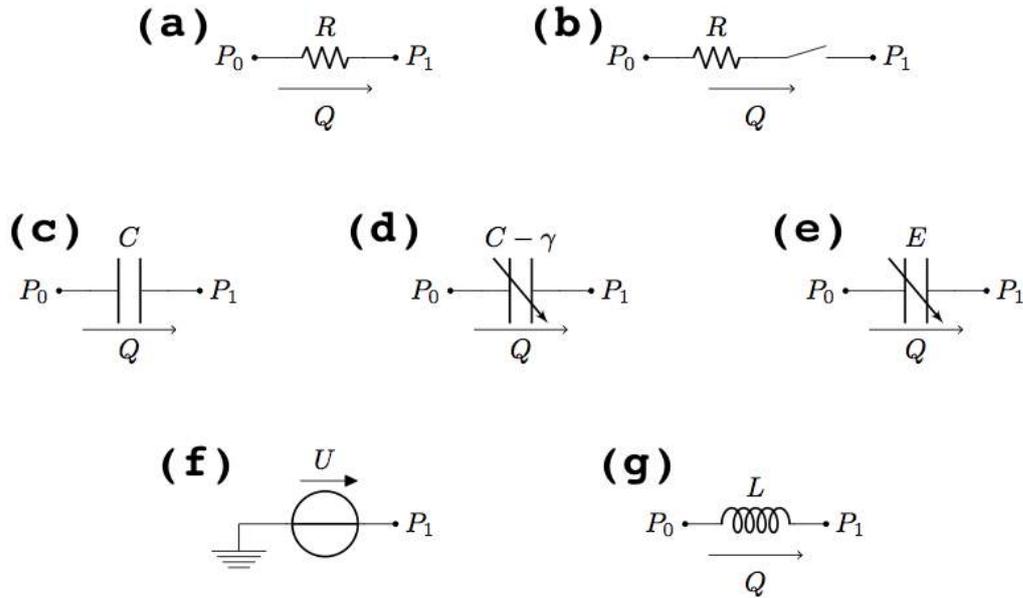


Figure 22.3 – Electrical elements included in the closed-loop model: (a) linear resistor; (b) linear resistor with ideal switch; (c) linear capacitor; (d) variable capacitor for arterial viscoelasticity; (e) variable capacitor for ventricular elastance; (f) voltage source; (g) linear inductor.

- *Linear resistor with ideal switch* (see Fig. 22.3(b)): in this element, often utilized to model heart valves [9, 73], the switch is completely open as soon as the pressure difference $P_0 - P_1$ is positive and completely closed otherwise, namely

$$Q = \sigma_{P_0 - P_1} \frac{P_0 - P_1}{R}, \quad (22.6b)$$

where $\sigma_{P_0 - P_1}$ is a binary-valued function of the pressure pair P_0, P_1 defined as:

$$\sigma_{P_0 - P_1} := \begin{cases} 1 & \text{if } P_0 - P_1 > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (22.6c)$$

- *Linear capacitor* (see Fig. 22.3(c)): in a capacitor, the time rate of change of the fluid volume V stored in the capacitor equals the volumetric flow rate Q , namely

$$\frac{dV}{dt} = Q. \quad (22.6d)$$

In the case of linear capacitor, the volume V and the pressure difference $P_0 - P_1$ are related by a proportionality law, namely

$$V = C(P_0 - P_1) \quad (22.6e)$$

where C is a positive constant.

- *Variable capacitor for arterial viscoelasticity* (see Fig. 22.3(d)): the law for the capacitor stated by Eq. (22.6d) is coupled with the following differential relationship between the pressure difference $P_0 - P_1$ and the fluid volume V

$$P_0 - P_1 = \frac{V}{C} + \gamma \frac{dV}{dt} \quad (22.6f)$$

where C and γ are positive constants. Relationship (22.6f) corresponds to adopting a linear viscoelastic thin shell model for the arterial walls, as in [29, 28].

- *Variable capacitor for ventricular elastance* (see Fig. 22.3(e)): the law for the capacitor stated by Eq. (22.6d) is coupled with the following relationship between the pressure difference $P_0 - P_1$ and the fluid volume V

$$P_0 - P_1 = E(t)V \quad (22.6g)$$

where $E(t)$ is a given function of time modeling the complex biomechanical properties of the ventricular wall. Following [9], we assume

$$E_L(t) = \text{ELD} + \text{ELS} a(t) \quad \text{and} \quad E_R(t) = \text{ERD} + \text{ERS} a(t)$$

where ELD, ELS, ERD and ERS are given constants and

$$a(t) = \begin{cases} \frac{1}{2} (\tanh(2\pi(T_m - T_a)) - \tanh(2\pi(T_m - T_b))) & \text{for } T_m < T_s \\ 0 & \text{otherwise.} \end{cases}$$

Here, T_m is defined as $T_m = \text{mod}(t, T_c)$, where T_s and T_c are the systolic and cardiac periods, respectively, and T_a and T_b are given constants.

- *Voltage source* (see Fig. 22.3(f)): the hydraulic analog of a voltage source is an element that imposes the nodal pressure as

$$P_1(t) = U(t) \quad (22.6h)$$

where $U(t)$ is a given function. Following [9], we assume

$$U_L(t) = \text{ULO} a(t) \quad \text{and} \quad U_R(t) = \text{URO} a(t)$$

where ULO and URO are positive constants and the function $a(t)$ is the same as described above.

- *Linear inductor* (see Fig. 22.3(f)): the time rate of change of the volumetric flow rate Q is related to the pressure difference $P_0 - P_1$ by the following proportionality law

$$L \frac{dQ}{dt} = P_0 - P_1 \quad (22.6i)$$

where L is a positive constant.

The parameter values for each of the circuit elements are reported in Table 22.1. The parameter values pertaining to the heart, the systemic micro-circulation and the pulmonary circulation have been adapted from [9]. The parameter values pertaining to the main arteries have been computed using the following constitutive equations

$$R = \frac{8\pi l \eta}{S^2} \quad L = \frac{\rho_b l}{S} \quad C = \frac{3lS(a+1)^2}{E(2a+1)} \quad \gamma = \frac{\delta}{C}$$

where $a = r/h$ is the ratio between vessel radius r and wall thickness h , l is the vessel length, $S = \pi r^2$ is the vessel cross-sectional area, ρ_b is the blood density, η is the blood viscosity, E and δ are the Young modulus and the viscoelastic parameter characterizing the vessel wall.

Here, we assume $\rho_b = 1.05 \text{ g cm}^{-3}$, $\eta = 0.035 \text{ g cm}^{-1} \text{ s}^{-1}$, $E = 4 \cdot 10^6 \text{ dyne cm}^{-2}$ and $\delta = 1.56 \cdot 10^{-3} \text{ s}$. The values of the remaining geometrical parameters utilized to determine R , L , C , C_v and γ for each of the main arterial segments have been adapted from [173] and [29] and are reported in Table 22.2.

HEART		
$T_c = 0.8$ s	$T_s = 0.4$ s	$T_a = 0.08$ s
$T_b = 0.45$ s	ULO = 50 mmHg	ELD = 0.04 mmHg cm ⁻³
ELS = 1.375 mmHg cm ⁻³ s ⁻¹	$R_L = 0.008$ mmHg s cm ⁻³	URO = 24 mmHg
ERD = 0.01 mmHg cm ⁻³	ERS = 0.23 mmHg cm ⁻³ s ⁻¹	$R_R = 0.0175$ mmHg s cm ⁻³
SYSTEMIC CIRCULATION		
$R_1 = 0.003751$ mmHg s cm ⁻³	$R_{2a} = 0.00003111$ mmHg s cm ⁻³	$R_{2b} = 0.00003111$ mmHg s cm ⁻³
$R_{3a} = 0.00011683$ mmHg s cm ⁻³	$R_{3b} = 0.00011683$ mmHg s cm ⁻³	$R_{4a} = 0.00061427$ mmHg s cm ⁻³
$R_{4b} = 0.00061427$ mmHg s cm ⁻³	$R_{5a} = 0.00101868$ mmHg s cm ⁻³	$R_{5b} = 0.00101868$ mmHg s cm ⁻³
$R_{6a} = 0.002652974$ mmHg s cm ⁻³	$R_{6b} = 0.002652974$ mmHg s cm ⁻³	$R_7 = 0.35$ mmHg s cm ⁻³
$R_8 = 0.0675$ mmHg s cm ⁻³	$R_9 = 2.0$ mmHg s cm ⁻³	$R_{10} = 0.003751$ mmHg s cm ⁻³
$C_2 = 0.13853688$ cm ³ mmHg ⁻¹	$\gamma_2 = 0.00713074$ mmHg s cm ⁻³	$C_3 = 0.12078980$ cm ³ mmHg ⁻¹
$\gamma_3 = 0.08117842$ mmHg s cm ⁻³	$C_4 = 0.21968142$ cm ³ mmHg ⁻¹	$\gamma_4 = 0.00449683$ mmHg s cm ⁻³
$C_5 = 0.17355893$ cm ³ mmHg ⁻¹	$\gamma_5 = 0.00569184$ mmHg s cm ⁻³	$C_6 = 0.02062154$ cm ³ mmHg ⁻¹
$\gamma_6 = 0.04790476$ mmHg s cm ⁻³	$C_7 = 0.8$ cm ³ mmHg ⁻¹	$C_8 = 1.46$ cm ³ mmHg ⁻¹
$C_9 = 20.0$ cm ³ mmHg ⁻¹	$L_3 = 0.00113873$ mmHg s ² cm ⁻³	$L_4 = 0.00424581$ mmHg s ² cm ⁻³
$L_5 = 0.00551999$ mmHg s ² cm ⁻³	$L_6 = 0.00538022$ mmHg s ² cm ⁻³	$L_7 = 0.000225$ mmHg s ² cm ⁻³
$L_8 = 0.000225$ mmHg s ² cm ⁻³	$L_9 = 0.0036$ mmHg s ² cm ⁻³	
PULMONARY CIRCULATION		
$R_{11} = 0.003751$ mmHg s cm ⁻³	$R_{12} = 0.03376$ mmHg s cm ⁻³	$R_{13a} = 0.1013$ mmHg s cm ⁻³
$R_{13b} = 0.003751$ mmHg s cm ⁻³	$C_{11} = 0.09$ cm ³ mmHg ⁻¹	$C_{12} = 2.67$ cm ³ mmHg ⁻¹
$C_{13} = 46.7$ cm ³ mmHg ⁻¹	$L_{12} = 0.00075$ mmHg s ² cm ⁻³	$L_{13} = 0.00308$ mmHg s ² cm ⁻³
CEREBRAL CIRCULATION		
$R_{14a} = 0.03006924$ mmHg s cm ⁻³	$R_{14b} = 0.03006924$ mmHg s cm ⁻³	$R_{cap1} = 0.327$ mmHg s cm ⁻³
$R_{cap2} = 0.327$ mmHg s cm ⁻³	$R_{15a} = 0.327$ mmHg s cm ⁻³	$R_{15b} = 0.327$ mmHg s cm ⁻³
$C_{14} = 0.03790125$ cm ³ mmHg ⁻¹	$\gamma_{14} = 0.04110141$ mmHg s cm ⁻³	$C_{15} = 0.688$ cm ³ mmHg ⁻¹
$L_{14} = 0.03430149$ mmHg s ² cm ⁻³	$L_{cap} = 0.00424581$ mmHg s ² cm ⁻³	$L_{15} = 0.00424581$ mmHg s ² cm ⁻³

Table 22.1 – Parameter values for the closed-loop model

ARTERIAL SEGMENT	l [cm]	r [cm]	h [cm]
Ascending Aorta	4	1.44	0.158
Aortic Arch	5.9	1.25	0.139
Thoracic Aorta	15.6	0.96	0.117
Abdominal Aorta	15.9	0.85	0.105
Iliac artery	5.8	0.52	0.076
Carotid artery	20.8	0.39	0.064

Table 22.2 – Geometrical parameters for the main arterial segments

Let us define the vector $Y(t)$ of the circuit unknowns in Eq. (22.5) as the column vector

$$Y(t) = [\mathcal{V}(t); \mathcal{Q}(t)]^T \tag{22.7}$$

where the two row vectors \mathcal{V} and \mathcal{Q} are defined as

$$\begin{aligned} \mathcal{V}(t) = & [V_L(t), V_2(t), V_3(t), V_4(t), V_5(t), V_6(t), V_7(t), V_8(t), \\ & V_9(t), V_R(t), V_{11}(t), V_{12}(t), V_{13}(t), V_{14}(t), V_{15}(t)] \\ \mathcal{Q}(t) = & [Q_3(t), Q_4(t), Q_5(t), Q_6(t), Q_7(t), Q_8(t), \\ & Q_9(t), Q_{12}(t), Q_{13}(t), Q_{14}(t), Q_{15}(t)]. \end{aligned}$$

The symbols V_L and V_R denote the fluid volume stored in the variable capacitors for ventricular elastance characterized by E_L and E_R , respectively; the symbols V_i , $i = 2, \dots, 6$ and $i = 14$ denote the fluid volume stored in the variable capacitors for arterial viscoelasticity characterized by C_i and γ_i ; the symbols V_i , $i = 7, \dots, 9$ and $i = 15$ denote the fluid volume stored in the linear capacitors characterized by C_i ; the symbols Q_i , $i = 3, \dots, 9$ and $i = 12, \dots, 15$, denote the volumetric flow rate through the inductor characterized by L_i .

To derive the nonlinear system of ODEs (22.5) representing the mathematical model of blood circulation in the human body, the following three steps are taken:

1. Kirchoff's current laws (KCLs) are written for each of the 15 nodes marked on the circuit in Fig. 22.2;
2. Kirchoff's voltage laws (KVLs) are written for each of the 11 circuit branches containing an inductor;
3. the constitutive equations (22.6) are substituted in the KCLs and KVLs.

The system is solved using the initial conditions reported in Table 22.3 and simulations are run until a periodic solution is established.

VARIABLE	VALUE	VARIABLE	VALUE
V_L	71.2700 cm ³	Q_3	1.68 cm ³ s ⁻¹
V_2	73.5486 cm ³	Q_4	1.9961 cm ³ s ⁻¹
V_3	71.9746 cm ³	Q_5	1.1861 cm ³ s ⁻¹
V_4	71.9983 cm ³	Q_6	9.03697 cm ³ s ⁻¹
V_5	71.9327 cm ³	Q_7	17.8121 cm ³ s ⁻¹
V_6	72.2213 cm ³	Q_8	19.1462 cm ³ s ⁻¹
V_7	80.9077 cm ³	Q_9	67.359 cm ³ s ⁻¹
V_8	70.537 cm ³	Q_{12}	0.7861 cm ³ s ⁻¹
V_9	3.3268 cm ³	Q_{13}	23.83 cm ³ s ⁻¹
V_R	3.1638 cm ³	Q_{14}	0.5909 cm ³ s ⁻¹
V_{11}	9.6619 cm ³	Q_{15}	1.9961 cm ³ s ⁻¹
V_{12}	9.5419 cm ³		
V_{13}	6.0704 cm ³		
V_{14}	70.9869 cm ³		
V_{15}	3.3268 cm ³		

Table 22.3 – Initial conditions for the time-dependent model simulations

Overall, the differential system (22.5) includes $m = 26$ differential equations. The expressions of the nonzero entries of the matrices \mathcal{M} and \mathcal{A} as well as of the forcing vector b are reported below. Let $\tilde{R}_L := R_L + R_1 + R_{2a}$, $\tilde{R}_R := R_R + R_{11}$, $\tilde{R}_{cap} := R_{14b} + R_{cap1} + R_{cap2} + R_{15a}$ and $\tilde{L}_{cap} := L_{cap} + L_{15}$. The nonzero entries of \mathcal{M} are:

$$\begin{aligned} \mathcal{M}_{1,1} &= 1, & \mathcal{M}_{12} &= -\frac{\gamma_2}{\tilde{R}_L} \sigma_{P_1-P_2} \\ \mathcal{M}_{2,2} &= 1 + \frac{\gamma_2}{\tilde{R}_L} \sigma_{P_1-P_2} \\ \mathcal{M}_{i,i} &= 1 \quad i = 3, \dots, 15 \end{aligned}$$

$$\begin{aligned}
\mathcal{M}_{i,j} &= \gamma_j, \quad \mathcal{M}_{i,j+1} = -\gamma_{j+1}, \quad \mathcal{M}_{i,i} = -L_{j+1}, \\
i &= 16, \dots, 19, \quad j = i - 14, \\
\mathcal{M}_{20,6} &= \gamma_6, \quad \mathcal{M}_{20,20} = -L_7 \\
\mathcal{M}_{i,i} &= -L_k, \quad i = 21, \dots, 24, \quad k = i - 13 \\
\mathcal{M}_{25,3} &= \gamma_3, \quad \mathcal{M}_{25,14} = -\gamma_{14}, \quad \mathcal{M}_{25,25} = -L_{14} \\
\mathcal{M}_{26,14} &= \gamma_{14}, \quad \mathcal{M}_{26,26} = -\tilde{L}_{cap}.
\end{aligned}$$

The nonzero entries of \mathcal{A} are:

$$\begin{aligned}
\mathcal{A}_{1,1} &= -E_L \left(\frac{\sigma_{P_{13}-P_1}}{R_{13b}} + \frac{\sigma_{P_1-P_2}}{\tilde{R}_L} \right) \\
\mathcal{A}_{1,2} &= \frac{\sigma_{P_1-P_2}}{\tilde{R}_L C_2}, \quad \mathcal{A}_{1,13} = \frac{\sigma_{P_{13}-P_1}}{R_{13b} C_{13}} \\
\mathcal{A}_{2,1} &= E_L \frac{\sigma_{P_1-P_2}}{\tilde{R}_L}, \quad \mathcal{A}_{2,2} = -\frac{\sigma_{P_1-P_2}}{\tilde{R}_L C_2}, \quad \mathcal{A}_{2,16} = -1 \\
\mathcal{A}_{3,16} &= 1, \quad \mathcal{A}_{3,17} = -1, \quad \mathcal{A}_{3,25} = -1 \\
\mathcal{A}_{i,j} &= 1, \quad \mathcal{A}_{i,j+1} = -1 \quad i = 4, \dots, 8, \quad j = i + 13, \\
\mathcal{A}_{9,9} &= -\frac{1}{C_9} \left(\frac{\sigma_{P_9-P_{10}}}{R_{10}} + \frac{1}{R_{15b}} \right) \\
\mathcal{A}_{9,10} &= \frac{E_R}{R_{10}} \sigma_{P_9-P_{10}}, \quad \mathcal{A}_{9,15} = \frac{1}{R_{15b} C_{15}}, \quad \mathcal{A}_{9,22} = 1 \\
\mathcal{A}_{10,9} &= \frac{\sigma_{P_9-P_{10}}}{R_{10} C_9}, \quad \mathcal{A}_{10,10} = -E_R \left(\frac{\sigma_{P_9-P_{10}}}{R_{10}} + \frac{\sigma_{P_{10}-P_{11}}}{\tilde{R}_R} \right) \\
\mathcal{A}_{10,11} &= \frac{\sigma_{P_{10}-P_{11}}}{\tilde{R}_R C_{11}}, \quad \mathcal{A}_{11,10} = E_R \frac{\sigma_{P_{10}-P_{11}}}{\tilde{R}_R}, \\
\mathcal{A}_{11,11} &= -\frac{\sigma_{P_{10}-P_{11}}}{\tilde{R}_R C_{11}}, \quad \mathcal{A}_{11,23} = -1, \quad \mathcal{A}_{12,23} = 1, \\
\mathcal{A}_{12,24} &= -1, \quad \mathcal{A}_{13,1} = E_L \frac{\sigma_{P_{13}-P_1}}{R_{13b}}, \quad \mathcal{A}_{13,13} = -\frac{\sigma_{P_{13}-P_1}}{R_{13b} C_{13}}, \\
\mathcal{A}_{13,24} &= 1, \quad \mathcal{A}_{14,25} = 1, \quad \mathcal{A}_{14,26} = -1 \\
\mathcal{A}_{15,9} &= \frac{1}{R_{15b} C_9}, \quad \mathcal{A}_{15,15} = -\frac{1}{R_{15b} C_{15}}, \quad \mathcal{A}_{15,26} = 1 \\
\mathcal{A}_{i,j} &= -\frac{1}{C_j}, \quad \mathcal{A}_{i,j+1} = \frac{1}{C_{j+1}}, \quad \mathcal{A}_{i,i} = (R_{jb} + R_{j+1a}), \\
i &= 16, \dots, 19, \quad j = i - 14, \\
\mathcal{A}_{20,6} &= -\frac{1}{C_6}, \quad \mathcal{A}_{20,7} = \frac{1}{C_7}, \quad \mathcal{A}_{20,20} = (R_7 + R_{6b}) \\
\mathcal{A}_{i,j} &= -\frac{1}{C_j}, \quad \mathcal{A}_{i,j+1} = \frac{1}{C_{j+1}}, \quad \mathcal{A}_{i,i} = R_{j+1}, \\
i &= 21, 22, \quad j = i - 14, \\
\mathcal{A}_{23,11} &= -\frac{1}{C_{11}}, \quad \mathcal{A}_{23,12} = \frac{1}{C_{12}}, \quad \mathcal{A}_{23,23} = R_{12} \\
\mathcal{A}_{24,12} &= -\frac{1}{C_{12}}, \quad \mathcal{A}_{24,13} = \frac{1}{C_{13}}, \quad \mathcal{A}_{24,24} = R_{13a} \\
\mathcal{A}_{25,3} &= -\frac{1}{C_3}, \quad \mathcal{A}_{25,14} = \frac{1}{C_{14}}, \quad \mathcal{A}_{25,25} = R_{14a}
\end{aligned}$$

$$\mathcal{A}_{26,14} = -\frac{1}{C_{14}}, \quad \mathcal{A}_{26,15} = \frac{1}{C_{15}}, \quad \mathcal{A}_{26,26} = \tilde{R}_{cap}.$$

The nonzero entries of the forcing term b are:

$$\begin{aligned} b_1 &= -U_L \left(\frac{\sigma_{P_{13}-P_1}}{R_{13b}} + \frac{\sigma_{P_1-P_2}}{\tilde{R}_L} \right) \\ b_2 &= \frac{U_L}{\tilde{R}_L} \sigma_{P_1-P_2}, \quad b_9 = \frac{U_R}{R_{10}} \sigma_{P_9-P_{10}} \\ b_{10} &= -U_R \left(\frac{\sigma_{P_9-P_{10}}}{R_{10}} + \frac{\sigma_{P_{10}-P_{11}}}{\tilde{R}_R} \right) \\ b_{11} &= \frac{U_R}{\tilde{R}_R} \sigma_{P_{10}-P_{11}}, \quad b_{13} = \frac{U_L}{R_{13b}} \sigma_{P_{13}-P_1}. \end{aligned}$$

3 Results

A qualitative description of left ventricular function is provided by the plot in Fig. 22.4, which reports the volume-pressure relationship in the left ventricle during one cardiac cycle simulated via the closed-loop model. Simulation results show that the closed-loop model correctly captures the four basic phases of ventricular function: ventricular filling (phase a), isovolumetric contraction (phase b), ejection (phase c), and isovolumetric relaxation (phase d).

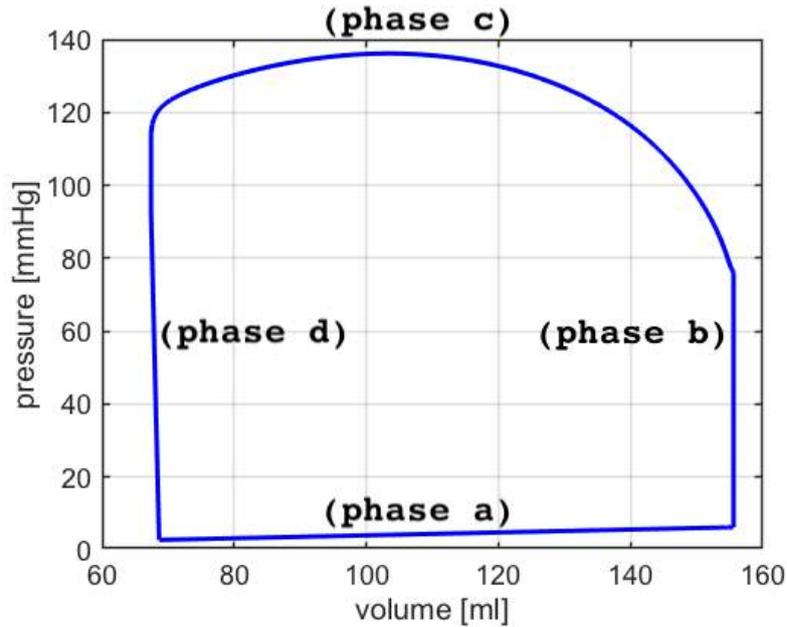


Figure 22.4 – Volume-pressure relationship in the left ventricle during one cardiac cycle simulated via the closed-loop model. Results show that the closed-loop model correctly captures the four basic phases of ventricular function: ventricular filling (phase a), isovolumetric contraction (phase b), ejection (phase c), and isovolumetric relaxation (phase d) [129].

Another qualitative representation of ventricular function is provided by the Wiggers diagram, where the volume waveform pertaining to the left ventricle is portrayed together with the pressure waveforms pertaining to the left ventricle and the ascending aorta. The Wiggers diagram simulated via the closed-loop model, reported in Fig. 22.5, shows the typical features of isovolumetric contraction and relaxation exhibited by physiological waveforms [129].

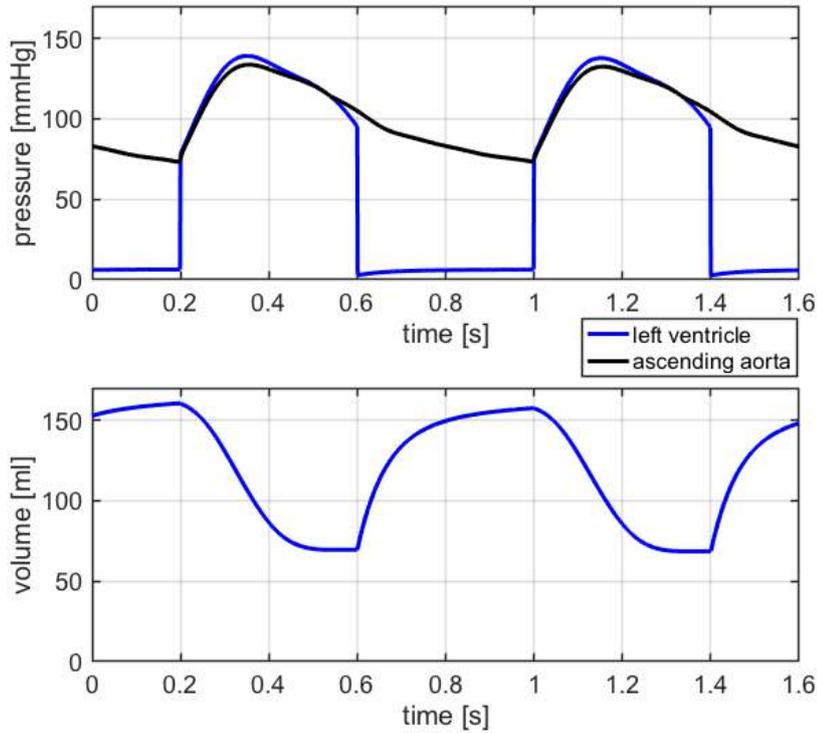


Figure 22.5 – Wiggers' diagram simulated via the closed-loop model. Results show the typical features of isovolumetric contraction and relaxation exhibited by physiological waveforms [129].

Quantitative parameters describing cardiovascular physiology include:

- end-diastolic volume EDV , which is the maximum value of the ventricular volume during the cardiac cycle;
- end-systolic volume ESV , which is the minimum value of the ventricular volume during the cardiac cycle;
- stroke volume $SV = EDV - ESV$;
- ejection fraction $EF = 100 \times \frac{SV}{EDV}$
- the length of the heart beat measured in seconds $T_c = \frac{60}{HR}$, where HR is the heart rate;
- cardiac output $CO = \frac{HR \times SV}{1000}$.

Tab. 22.4 reports the values of these parameters for the left and right ventricle as reported in the clinical literature and simulated via the closed-loop model. Specifically, the clinical studies

in [146] and [147] utilized cardiovascular magnetic resonance to assess left and right ventricular functions on 120 healthy individuals. All the simulated values fall within the ranges reported in the clinical literature, thereby validating the capability of the closed-loop model to capture the main features of the heart functions.

PARAMETER	UNIT	NORMAL CLINICAL RANGE		CLOSED-LOOP MODEL	
		<i>Left Ventricle</i>	<i>Right Ventricle</i>	<i>Left Ventricle</i>	<i>Right Ventricle</i>
End-Diastolic Volume (EDV)	[ml]	142 (102,183) [146]	144 (98,190) [147] 100 - 160 [141]	157.5	154.1
End-Systolic Volume (ESV)	[ml]	47 (27,68) [146]	50 (22,78) [147] 50-100 [141]	68.6	65.0
Stroke Volume (SV)	[ml/beat]	95 (67, 123) [146] 60 - 100 [141]	94 (64, 124) [147] 60-100 [141]	88.9	89.1
Cardiac Output (CO)	[l/min]	4-8 [141]	4-8 [141]	6.7	6.7
Ejection Fraction (EF)	[%]	67 (58, 76) [146]	66 (54, 78) [147] 40 - 60 [141]	56.5	57.8

Table 22.4 – Cardiovascular Physiology: Left and Right Ventricles

The pressure and volume waveforms pertaining to the main segments of the systemic arteries simulated via the closed-loop model are reported in Fig. 22.2. Results show that the model captures the typical features of peak magnification and time delay exhibited by physiological waveforms [157]. In Fig. 22.6, the simulated pressure waveforms are compared with experimental measurements at different sites along the arterial tree. In particular, we compare the results of our simulations with those presented by

- (a) Davies et al [55, 56], where sensor-tipped intra-arterial wires are used to measure pressure in nearly 20 subjects (age, 35-73 years) at 10-cm intervals along the aorta, starting at the aortic root;
- (b) Parker [179], where aortic pressures at 10, 20, 30, 40, 50 and 60 cm downstream from the aortic valve are reported;
- (c) Epstein et al [66], where arterial waveforms are generated via a one-dimensional arterial network.

For the ease of comparison, all the waveforms have been normalized to a unitary amplitude and have been rescaled in time to last 0.8 s. Fig. 22.6 shows a very good agreement between theoretical and experimental waveforms.

Overall, the results presented in this section provide evidence of the capability of proposed closed-loop model to capture prominent features of cardiovascular physiology and support the utilization of the model output to construct a theoretical BCG.

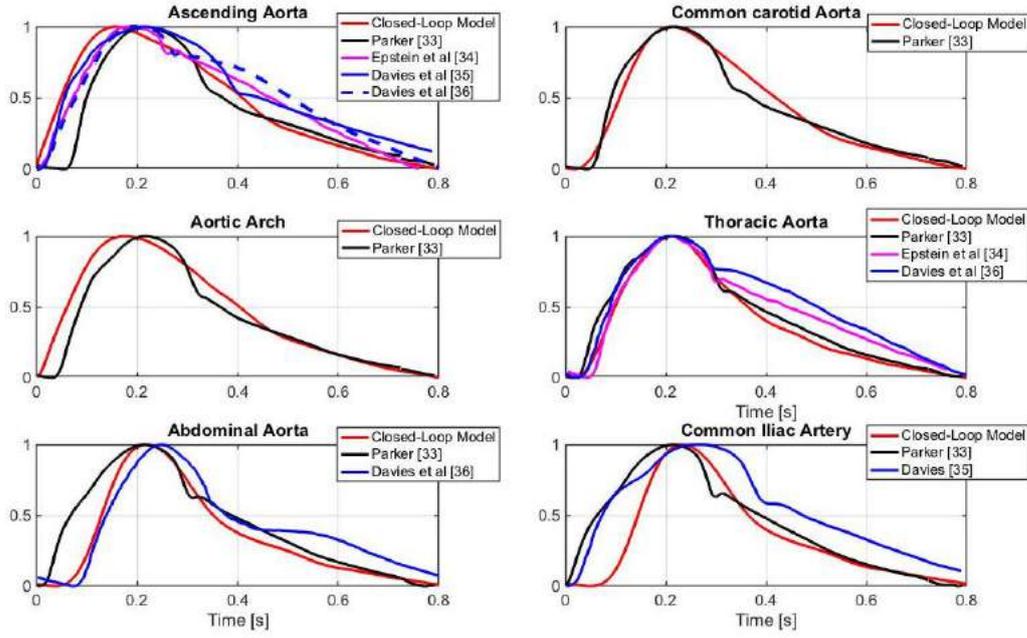


Figure 22.6 – Pressure waveforms simulated via the proposed closed-loop model (*red*) are compared with (i) experimental measurements at different sites along the arterial tree reported by Parker [179] (*black*) and Davies et al [55, 56] (*blue*); and (ii) theoretical simulations generated via a one-dimensional arterial network by Epstein et al [66] (*magenta*).

4 Theoretical and experimental ballistocardiograms

The volume waveforms simulated using the closed-loop model (see Fig. 22.2) have been utilized in Eq. (22.2) to calculate theoretically the waveform $BCG_{disp}(t)$ associated with the displacement of the center of mass. The BCG waveforms for velocity and acceleration, namely $BCG_{vel}(t)$ and $BCG_{acc}(t)$, are obtained from $BCG_{disp}(t)$ via discrete time-differentiation.

Following [173, 171, 229], the BCG waveforms are reported by means of the auxiliary functions f_D , f_V and f_A defined as

$$\begin{aligned}
 f_D(t) &:= M \cdot BCG_{disp}(t) = \rho_b \sum_{i=1}^N V_i(t) y_i && [\text{g cm}] \\
 f_V(t) &:= M \cdot BCG_{vel}(t) = \rho_b \sum_{i=1}^N \frac{dV_i(t)}{dt} y_i && [\text{g cm s}^{-1}] \\
 f_A(t) &:= M \cdot BCG_{acc}(t) = \rho_b \sum_{i=1}^N \frac{d^2V_i(t)}{dt^2} y_i && [\text{dyne}].
 \end{aligned}$$

These auxiliary functions are independent of the value of mass M and, as a consequence, they allow for a fair comparison between BCG waveforms reported in different studies.

Fig. 22.7 reports the waveform $f_A(t)$ simulated via the closed-loop model over one cardiac cycle. The waveform exhibits the typical I, J, K, L, M and N peaks and valleys that characterize BCG signals measured experimentally [229, 126, 113], thereby confirming the capability of the closed-loop model to capture the fundamental cardiovascular mechanisms that give rise to the BCG signal.

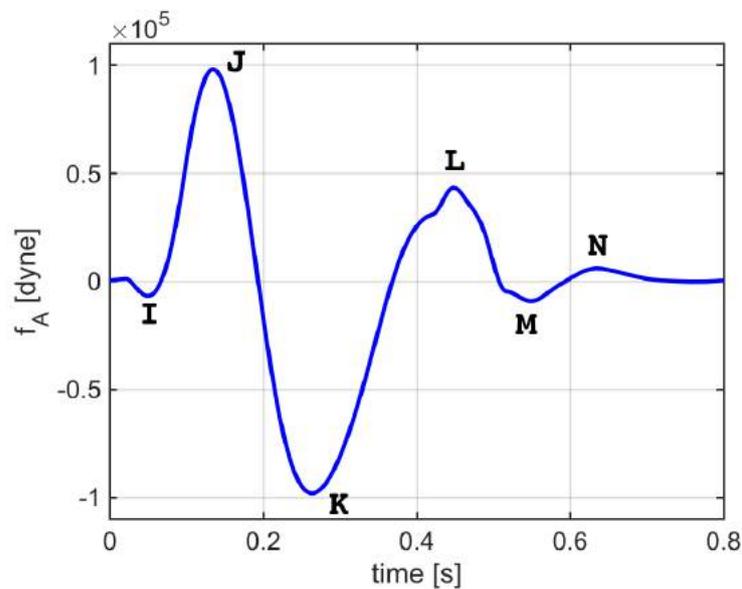


Figure 22.7 – Waveform $f_A(t)$ associated with the acceleration of the body motion simulated via the closed-loop model over one cardiac cycle. The waveform exhibits the typical I, J, K, L, M and N peaks that characterize BCG signals measured experimentally.

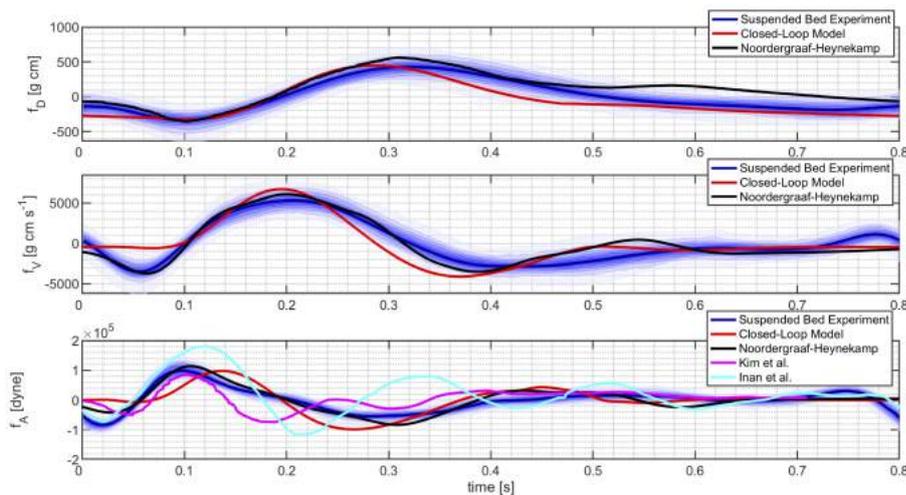


Figure 22.8 – BCG waveforms for displacement, velocity and acceleration, denoted by f_D , f_V and f_A , respectively. The figure reports the waveforms simulated via the closed-loop model, the theoretical waveforms calculated by Noordergraaf and Heynekamp [171] and the experimental waveforms obtained by Inan et al [113], Kim et al [126] and our group.

A quantitative comparison between the simulated BCG waveforms pertaining to displacement, velocity and acceleration of the center of mass is reported in Fig. 22.8 by means of the auxiliary functions f_D , f_V and f_A .

The waveforms simulated via the closed-loop model are compared with

1. the theoretical results presented by Noordergraaf and Heynekamp [171], where we calculated velocity and acceleration waveforms via time-differentiation of the reported signal for the displacement;

2. the experimental results obtained by Inan et al utilizing a modified weighing scale [113, 112];
3. the experimental results obtained by Kim et al utilizing direct pressure measurements [126];
4. the experimental results obtained by our group utilizing an accelerometer on a suspended bed (see [94]), where we utilized the mass $M = 87$ Kg of the study subject to calculate f_A .

A time-shift has been applied in order to align the waveforms for ease of comparison.

The results in Fig. 22.8 confirm that the BCG waveforms simulated via the closed-loop model are of the same order of magnitude as those reported in previous theoretical and experimental studies for all three aspects of the body motion, namely displacement, velocity and acceleration. The figure also shows that theoretical and experimental BCG waveforms obtained by different authors exhibit similar trends, typically marked by the peaks reported in Fig. 22.7, but differ in the precise timing of these peaks and in their magnitude. Such differences might be due to physiological variability of cardiovascular parameters among individuals, for example associated with age, gender and ethnicity [146, 147, 32], and to different techniques utilized to acquire the BCG signal, such as the modified weighing scale and the accelerometer on the suspended bed.

5 Conclusions

The closed-loop model presented in this chapter reproduces the predominant features of the physiology of the human cardiovascular system that give rise to the BCG signal. To the best of our knowledge, this is the first theoretical interpretation of the BCG signal based on a biophysical, mathematical closed-loop model of the cardiovascular system. A validation is provided with comparison against actual measurements, matching not just qualitatively but also quantitatively.

A limitation of the proposed closed-loop model is that it includes only a few of the main arteries that may have a significant contribution to the BCG signal [229, 171].

The idea of this project is to have a preliminary analysis using the proposed BCG model to aid in the tracking of a person's cardiovascular health noninvasively by examining the morphology of the BCG signal. Using a model of the BCG sensing system, the output signal can be translated to the standard base signal for study against the simulated model output. That is, the closed-loop model will be used to simulate pathological conditions affecting the cardiovascular system, such as hypertension or congestive heart failure, and the results will be compared to the measured BCG signals. Thus, with continued effort, there is a significant potential in better monitoring of cardiovascular health, particularly important for our aging population.

Electrostatic simulation of a floating-gate nMOS transistor in inversion conditions

Contents

1	Electronic model	247
2	Data	249
3	Results	251
4	Conclusions	253

In this chapter we study the distribution of the electric potential V in a nanoscale floating-gate nMOS (Metal-Oxide-Semiconductor) transistor working in inversion conditions. The prefix “n” indicates that the electric current is due to negatively charged electrons. This is an example of application in electronics of the HDG method with IBC presented in Ch. 8. This work has been completed in collaboration with Riccardo Sacco, professor at the *Politecnico di Milano*.

1 Electronic model

Under the assumption of electro-neutrality, the distribution of V is obtained as the solution of a linear elliptic problem that can be written in mixed form similarly to Eq. (4.1), see [217, Section 6.1], [152, Sections 4.6-4.7] and [163, Section 9.1]) as:

$$\underline{D} + \varepsilon \nabla V = 0 \quad \text{in } \Omega, \quad (23.1a)$$

$$\nabla \cdot \underline{D} = 0 \quad \text{in } \Omega. \quad (23.1b)$$

Comparing (23.1) with (4.1), we see that \underline{j} corresponds to the electric displacement \underline{D} , p corresponds to the electric potential V and \underline{K} is equal to the identity matrix times the dielectric

permittivity ε . According to the theory of Maxwell equations [116], the electric displacement \underline{D} is related to the electric field $\underline{E} = -\nabla V$ through the constitutive equation

$$\underline{D} = \varepsilon \underline{E} \quad \text{in } \Omega. \quad (23.1c)$$

Thus, using a finite element procedure, once the displacement \underline{D} is computed, the electric field can be easily determined on each $K \in \mathcal{T}_h$ by inverting the algebraic equation (23.1c) instead of being post-processed through numerical differentiation of the electric potential V as in standard displacement-based finite element formulations.

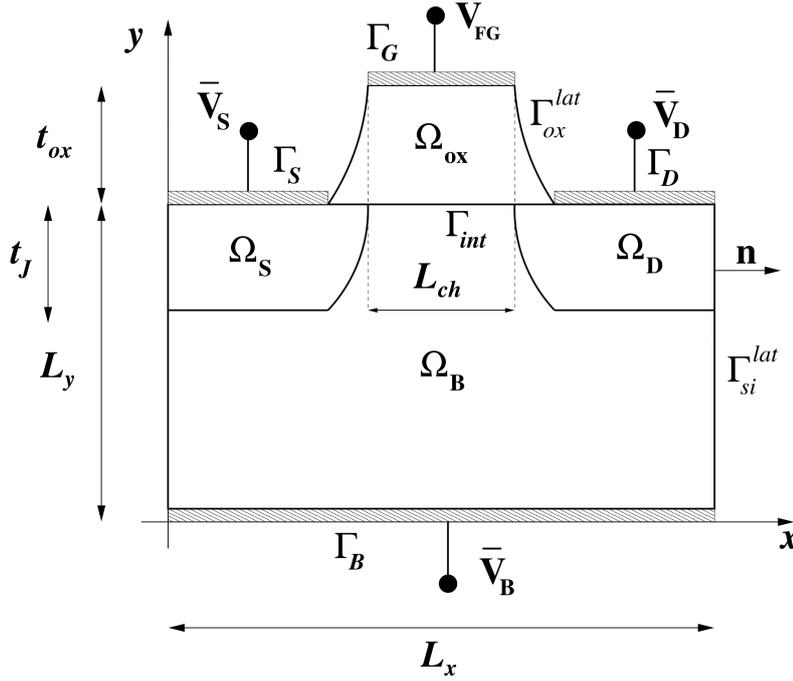


Figure 23.1 – Schematic representation of the two-dimensional cross-section of a nMOS transistor.

The computational domain Ω is the three-dimensional bounded set obtained by prolongating the scheme of Figure 23.1 by a width L_z along the axis z perpendicular to the plane of the figure. The domain Ω is the union of four sub-domains, namely Ω_{ox} (silicon dioxide, here simply *oxide*), Ω_S (*source*), Ω_D (*drain*) and Ω_B (*bulk*). The union of the last three sub-domains represents the silicon semiconductor region, namely $\Omega_{si} = \Omega_S \cup \Omega_D \cup \Omega_B$. We remark that the dielectric permittivity ε appearing in equations (23.1a) and (23.1c) can be written as $\varepsilon = \varepsilon_0 \varepsilon_r$, where ε_0 is the dielectric permittivity of vacuum and ε_r is the relative dielectric permittivity of silicon and oxide, equal to ε_r^{si} in Ω_{si} and ε_r^{ox} in Ω_{ox} , respectively. The sub-domains, Ω_{si} and Ω_{ox} are separated by an interface surface $\Gamma_{int} = \partial\Omega_{si} \cap \partial\Omega_{ox}$. The electric charge transport occurs only in the semiconductor region whereas the oxide is assumed to be a perfect insulator. Figure 23.1 also reports the notation for the various domain boundaries and their geometrical dimensions.

Equations (23.1) are equipped with the following boundary conditions:

$$V = \bar{V}_j = \bar{V}_{bi,j} \quad \text{on } \Gamma_j, j = S, D, B, \quad (23.2a)$$

$$\underline{D} \cdot \underline{n} = 0 \quad \text{on } \Gamma_{si}^{lat} \cup \Gamma_{ox}^{lat}, \quad (23.2b)$$

$$\int_{\Gamma_G} \underline{D} \cdot \underline{n} d\Sigma = q \bar{N}_B \delta L_{ch} L_z \quad \text{on } \Gamma_G, \quad (23.2c)$$

and interface conditions:

$$[[V]]_{\Gamma_{\text{int}}} = 0 \quad \text{on } \Gamma_{\text{int}}, \quad (23.3a)$$

$$[[D]]_{\Gamma_{\text{int}}} = q\bar{N}_B\delta \quad \text{on } \Gamma_{\text{int}}. \quad (23.3b)$$

The quantity q is the electron charge, \bar{N}_B is the concentration of ionized dopants in the bulk region and δ is the width of the accumulation region in the y direction of the channel. This quantity can be estimated using a simplified one-dimensional theory from solid state physics (see [163, Section 9.1]) as

$$\delta = \sqrt{\frac{4\varepsilon_0\varepsilon_r^{\text{si}}V_{\text{th}}\ln(\bar{N}_B/n_i)}{q\bar{N}_B}}, \quad (23.4)$$

where n_i is the silicon intrinsic concentration and V_{th} is the thermal voltage (see [217, Ch. 2]).

The Dirichlet boundary conditions (23.2a) express the physical fact that the electrical contacts are equipotential surfaces. The homogeneous Neumann boundary condition (23.2b) expresses the physical assumption that the nMOS device does not exchange electrical energy with the surrounding environment. The nonhomogeneous integral condition (23.2c) expresses the physical fact that the oxide layer acts like a capacitor because the right-hand side of (23.2c) is the total positive charge that is attracted on the contact from the external connecting wire to counterbalance the negative charge that is accumulated under the surface separating the silicon substrate from the oxide layer when $V_{FG} = V_T$.

The interface conditions (23.3a)- (23.3b) express the compatibility relations for the normal component of the displacement vector and the traces of the electric potential across an interface separating two different dielectric media [116]. We notice that (23.3a) expresses the physical fact that the electric potential is a continuous function when passing across the interface whereas (23.3b) expresses the physical fact that the jump in the normal component of the displacement vector is balanced by the total negative charge attracted from the substrate by the floating gate to form the n -type channel. This agrees with the integral condition (23.2c) because the two charge sheets distributed uniformly on Γ_{int} and on Γ_G are equal in magnitude and opposite in sign consistently with the fact that the oxide layer functions as a (nonlinear) capacitor whose capacitance depends on the biasing conditions of the device.

2 Data

Table 23.1 reports the numerical values of the physical parameters, whereas Table 23.2 provides a list of the input data used for device simulation. The quantities \bar{N}_S and \bar{N}_D are the concentrations of the ionized dopant impurities in the source and drain regions, respectively. With the data reported in Tables 23.1 and 23.2, the value of the width of the accumulation region δ is of about 16.3nm. This value is sufficiently smaller than the characteristic size of the device $\sqrt{L_x^2 + L_y^2 + L_z^2} = 660\text{nm}$ to make the assumption of semiconductor electro-neutrality a very reasonable one.

When the nMOS transistor is in the “off” state, the source and drain regions are electrically disconnected and no current flows in the device. To switch the transistor in the “on” state we need to allow electron flow from source to drain. This is obtained by applying suitable external

Parameter	Symbol	Value	Units
Electron charge	q	$1.602 \cdot 10^{-19}$	C
Permittivity of vacuum	ε_0	$8.854 \cdot 10^{-12}$	F m ⁻¹
Relative permittivity of silicon	$\varepsilon_r^{\text{si}}$	11.7	[-]
Relative permittivity of silicon dioxide	$\varepsilon_r^{\text{ox}}$	3.9	[-]
Thermal voltage (at $T = 300\text{K}$)	V_{th}	0.02589	V
Intrinsic concentration (at $T = 300\text{K}$)	n_i	$1.45 \cdot 10^{16}$	m ⁻³

Table 23.1 – Values of physical parameters. Data from [163].

Parameter	Symbol	Value	Units
Horizontal length	L_x	$480 \cdot 10^{-9}$	m
Vertical length	L_y	$320 \cdot 10^{-9}$	m
Width	L_z	$320 \cdot 10^{-9}$	m
Oxide thickness	t_{ox}	$10 \cdot 10^{-9}$	m
Source and drain lengths	$ \Gamma_S , \Gamma_D $	$160 \cdot 10^{-9}$	m
Channel length	L_{ch}	$40 \cdot 10^{-9}$	m
Junction depth	t_J	$106 \cdot 10^{-9}$	m
Source potential	\bar{V}_S	0	V
Bulk potential	\bar{V}_B	0	V
Drain potential	\bar{V}_D	0	V
Source doping concentration	\bar{N}_S	10^{26}	m ⁻³
Bulk doping concentration	\bar{N}_B	$5 \cdot 10^{22}$	m ⁻³
Drain doping concentration	\bar{N}_D	10^{26}	m ⁻³

Table 23.2 – Values of input parameters for device simulation.

voltages at the electrical contacts Γ_S (source contact), Γ_D (drain contact), Γ_B (bulk contact) and Γ_G (floating gate contact). We denote by $\bar{V}_{a,j}$, $j = S, D, B$, the given external voltages applied to each contact, and by $\bar{V}_{bi,j}$, $j = S, D, B$, the "built-in" potential of each contact. Each built-in potential $\bar{V}_{bi,j}$ is a function of the concentration of the fixed ionized dopant impurities that are introduced in the corresponding semiconductor region Ω_j to increase the electrical conductivity of the material. We refer to [87] for the technological process and to [217, Ch. 2] for the mathematical description. Finally, we denote by $\bar{V}_j = \bar{V}_{a,j} + \bar{V}_{bi,j}$, $j = S, D, B$, the boundary values of the electric potential V at the source, drain and bulk contacts, respectively, whereas V_{FG} is the value of the electric potential at the floating gate.

In the present study, we consider the case where source, drain and bulk contacts are grounded, corresponding to set $\bar{V}_{a,j} = 0$, $j = S, D, B$, and we aim at determining the value of V_{FG} at which the floating gate must be set in such a way that the electron concentration in the channel underneath Γ_{int} equals the dopant concentration in the bulk region. In such electrical configuration the nMOS works in *inversion condition* and the value of the floating gate potential, V_T , takes the name of *threshold voltage* (see [163]). The threshold voltage is a technologically fundamental parameter because if $V_{FG} < V_T$ no "channel" exists between source and drain and therefore no current flows - corresponding to no self-heating of the device; instead, if $V_{FG} = V_T$, a "channel" forms to connect source and drain regions and by furtherly applying a voltage difference between Γ_D and Γ_S , electrons are injected from the source to the drain region giving rise to electric current (corresponding to self-heating of the device). Thus, an accurate prediction of V_T allows to accurately control the thermal dissipated power of an electronic component [200].

3 Results

In this section we present the simulation results conducted using the HDG formulation presented in Ch. 8 using the same computational framework FEEL++ (Ch. 9).

The main goal of the simulations is to determine the value of V on Γ_G , denoted by V_{FG} via the integral boundary condition (23.2c). The value of V_{FG} at the floating gate is also known as threshold voltage of the device and is a fundamental design parameter in integrated circuit nanoelectronics (see [163]).

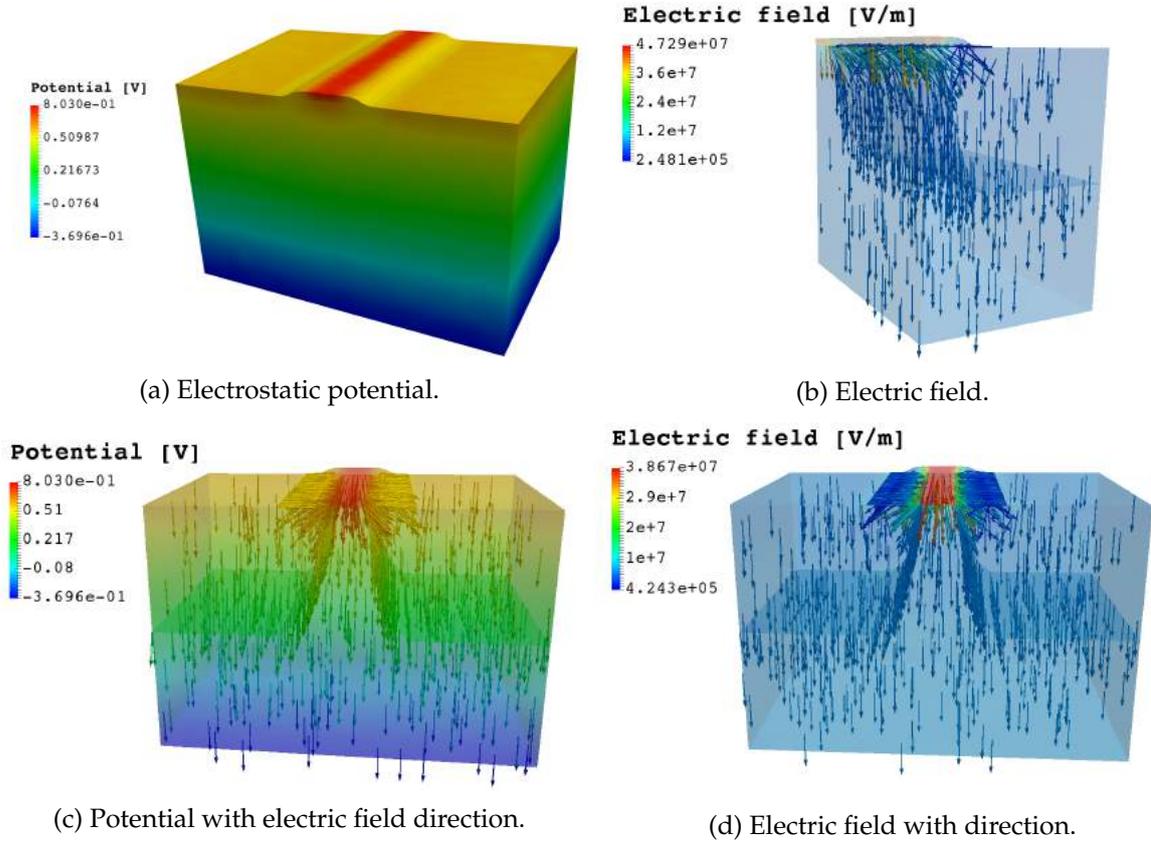


Figure 23.2 – 3D view of significant simulated quantities.

Figure 23.2a shows the 3D color plot of the computed electric potential V whereas Figure 23.2b illustrates a 3D vector plot of the computed electric field \mathbf{E} represented on a vertical cut of the device geometry in correspondence with the central section at $x = L_x/2$. We notice that the electric potential is an increasing function of the vertical coordinate y , negative at the bulk contact and positive at the source and drain contacts. The value of V at the floating gate, which is the threshold voltage of the device, computed using the integral boundary condition (23.2c), is positive and given by $V_T^{\text{computed}} = 0.8337965$ V. The sign of V_T^{computed} is in physical agreement with the fact that the transistor is of n type, so that electron charges (negative) are attracted from the bulk region up towards the gate contact. To quantitatively assess the accuracy of the estimate of the threshold voltage we adopt the analytical theory for an ideal MOS system developed in [163, Chapter 8] and use formula (8.3.18) of [163] to obtain $V_T^{\text{ideal}} = 0.8591$, which agrees with the value predicted by the numerical simulation within 3%.

Figure 23.2c illustrates the 3D distribution of the computed electric potential superposed with the vector plot of the electric field. It is remarkable to notice that this latter quantity is always directed from the gate of the device towards the bulk contact of the device in such a way to attract electron charge from the bulk region towards the interface Γ_{int} to form the conductive channel. We can also notice the separation lines between the source and bulk regions and the drain and bulk regions, which correspond to a change of the sign of the ionized dopant charge density - positive in Ω_S and Ω_D , negative in Ω_B . Figure 23.2d shows a vector plot of the computed electric field with a color map of its strength throughout the device. Results clearly indicate that the electric field is mainly directed parallel to the y axis except in the region comprising the oxide layer, the interface and the two source and drain contacts, where the electric field arrows strongly deviate from the y direction and point from the floating gate towards the two contacts, in accordance with the fact that the electric potential is much higher on Γ_G than on Γ_S and Γ_D .

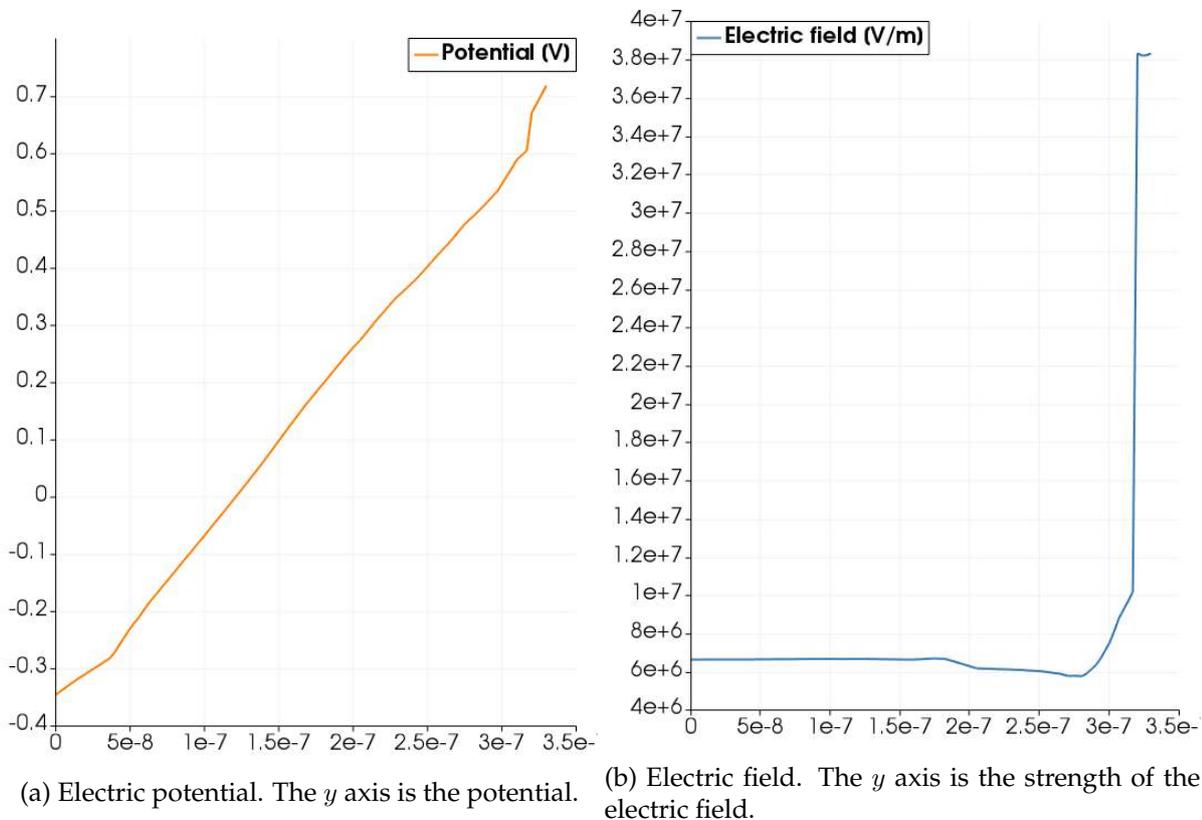


Figure 23.3 – 1D cut of significant electric variables at $x = L_x/2$.

Figure 23.3a shows a 1D cut of the computed electric potential at $x = L_x/2$ along the y axis. We notice that the spatial distribution of V is approximately linear in both semiconductor and oxide regions, with a change in the slope across the interface coordinate $y = L_y = 320\text{nm}$. This is physically consistent with the fact that

- (a) the relative dielectric constant ϵ_r has a jump passing from silicon to oxide;
- (b) there is no space charge density inside each region of the device (the right-hand side of the Poisson equation (23.1b)).

To further support these considerations about the electric potential V , we show in Figure 23.3b the 1D cut of the computed electric field strength at $x = L_x/2$ along the y axis. We notice that the field is discontinuous across the interface, in agreement with (a), and that it is approximately constant in both semiconductor and oxide regions, in agreement with (b).

4 Conclusions

We have just presented an application in semiconductor design of the innovative method HDG with IBC we have theorized and described in Ch. 8. This advanced numerical method allowed us a considerable gain in computational cost and robustness to compute without the need of sub-iterations or trial-and-error technique the value of V_{FG} , crucial for the design of a nMOS transistor.

This application can be extended in order to design a full electronic circuit with a 3D zoom on the transistor exploiting also the 3D-0D coupling illustrated in Ch. 7. An example of such circuit, which may be relevant for future applications in industry, is depicted in Fig. 23.4.

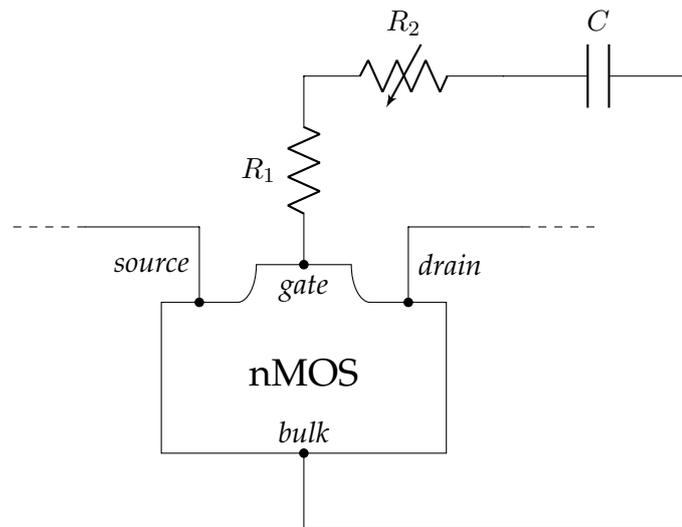


Figure 23.4 – 3D-0D electronic circuit in view of an industrial application.

Contents

1	Data	255
2	Results	257
2.1	Prism test case	257
2.2	Cylinder test case	259
3	Conclusions	261

In this chapter we exploit the HDG formulation with IBC for linear elasticity equations (Eq. (5.1)) to perform a virtual traction test. The idea of a traction test for a material is to pull or push a bar or small cylinder of this material with a specific force to retrieve the displacement and therefore compute the material properties. In particular we focus on the uniaxial test (Fig. 24.1)

1 Data

The traction test is simulated in two different domains: prism (Fig. 24.2a) and cylinder (Fig. 24.2b). These computational domains are defined by the following geometric description

$$\begin{aligned} \Omega_{prism} &= \{ \underline{x} \in \mathbb{R}^3 : x_1 \in (0, a), x_2 \in (0, a), x_3 \in (0, L) \} \\ \Gamma_{0,prism} &= \{ \underline{x} \in \mathbb{R}^3 : x_1 \in [0, a], x_2 \in [0, a], x_3 = 0 \} \\ \Gamma_{lat,prism} &= \{ \underline{x} \in \mathbb{R}^3 : x_1 \in \{0, a\}, x_2 \in \{0, a\}, x_3 \in (0, L) \} \\ \Gamma_{L,prism} &= \{ \underline{x} \in \mathbb{R}^3 : x_1 \in [0, a], x_2 \in [0, a], x_3 = L \} \\ \Omega_{cylinder} &= \{ \underline{x} \in \mathbb{R}^3 : x_1^2 + x_2^2 < R^2, x_3 \in (0, L) \} \end{aligned}$$

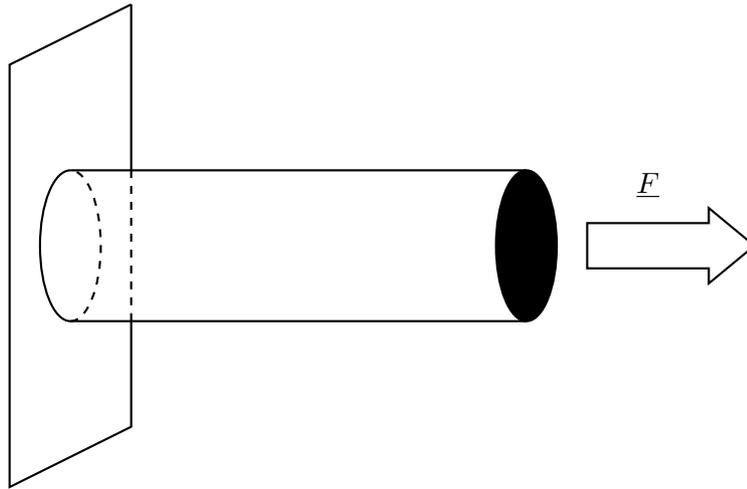


Figure 24.1 – Uniaxial test

$$\begin{aligned} \Gamma_{0,cyl} &= \{ \underline{x} \in \mathbb{R}^3 : x_1^2 + x_2^2 \leq R^2, x_3 = 0 \} \\ \Gamma_{lat,cyl} &= \{ \underline{x} \in \mathbb{R}^3 : x_1^2 + x_2^2 = R^2, x_3 \in (0, L) \} \\ \Gamma_{L,cyl} &= \{ \underline{x} \in \mathbb{R}^3 : x_1^2 + x_2^2 \leq R^2, x_3 = L \} \end{aligned}$$

with

$$a = 2 \text{ cm}$$

$$L = 5 \text{ cm}$$

$$R = 1 \text{ cm}$$

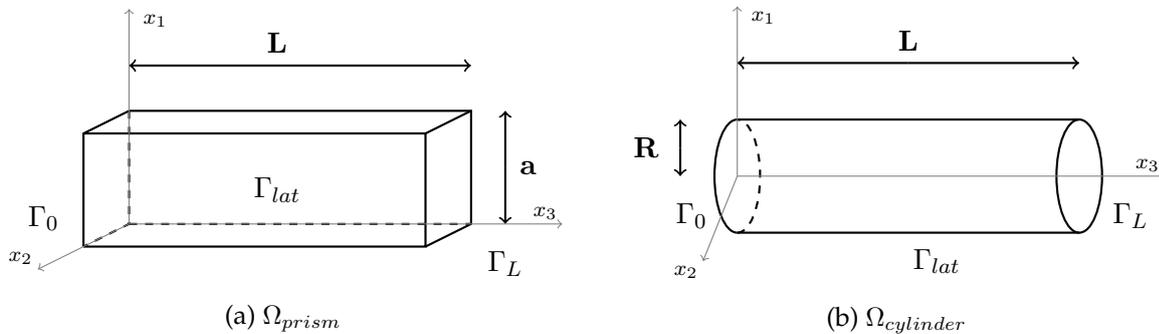


Figure 24.2 – Computational domains for the traction test.

For what concerns the model parameters data and boundary conditions we have

$$\begin{aligned} \lambda &= 0.115 & \mu &= 0.077 \\ \underline{u} &= \underline{0} \text{ on } \Gamma_0 & \underline{\underline{\sigma}} \cdot \underline{n} &= \underline{0} \text{ on } \Gamma_{lat} \end{aligned}$$

For the boundary condition on Γ_L we compare two different simulation, one with Neumann boundary condition (NBC), the other with IBC, in particular:

- NBC: $\underline{\underline{\sigma}} \cdot \underline{n} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$ on Γ_L

- **IBC:** $\int_{\Gamma_L} \underline{\underline{\sigma}} \cdot \underline{\underline{n}} = \begin{bmatrix} 0 \\ 0 \\ g |\Gamma_L| \end{bmatrix}$ on Γ_L

with $g = 0.8$.

2 Results

In this section we compare the results obtained with the HDG linear elasticity class developed in FEEL++. We compare the simulations using Neumann and integral boundary condition in the case of a prism, first, and a cylinder, then.

2.1 Prism test case

Let us begin by assessing the capability of the proposal algorithm in imposing the desired boundary condition: $\underline{\underline{\sigma}} \cdot \underline{\underline{n}}$ and $\int_{\Gamma} \underline{\underline{\sigma}} \cdot \underline{\underline{n}}$. From the results presented in Tab. 24.1 we can infer that the algorithm allows to impose the desired condition with good accuracy in both NBC and IBC test cases.

NBC	IBC
$\mathbf{F}_{NBC} = \begin{bmatrix} 6.82 \cdot 10^{-9} \\ 7.917494 \cdot 10^{-9} \\ 3.20 \end{bmatrix}$	$\mathbf{F}_{IBC} = \begin{bmatrix} -1.188909 \cdot 10^{-5} \\ -4.200072 \cdot 10^{-5} \\ 3.199062 \end{bmatrix}$

Table 24.1 – Table of results for the prism test case.

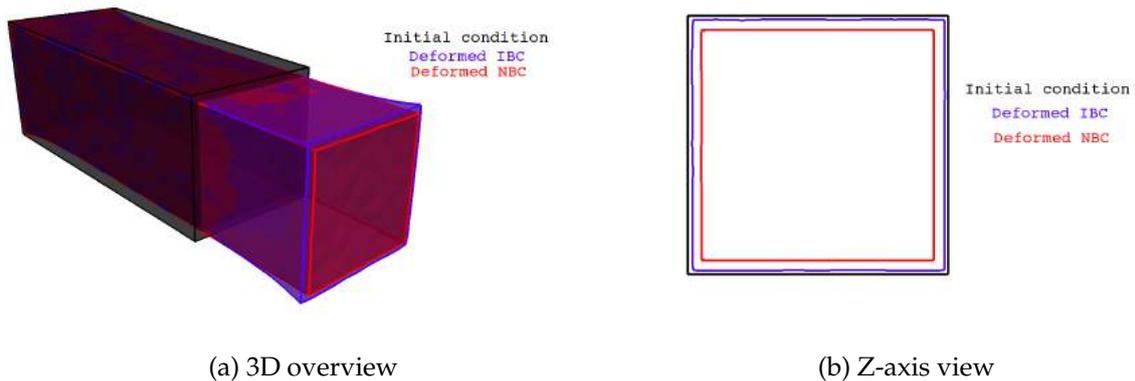


Figure 24.3 – Initial and final condition for the prism test case.

Let now investigate the differences between the solution of the NBC and IBC test cases from the physical viewpoint. Looking at Fig. 24.3a, we observed that the two deformed configuration have been stretched qualitatively by the same amount along the main direction. On the other side we can see that there are little differences on the surface of interest Γ_L , indeed if we compare the initial condition of the surface and the two final deformed configuration as shown in Fig. 24.3b, we notice a wider surface for IBC than for NBC.

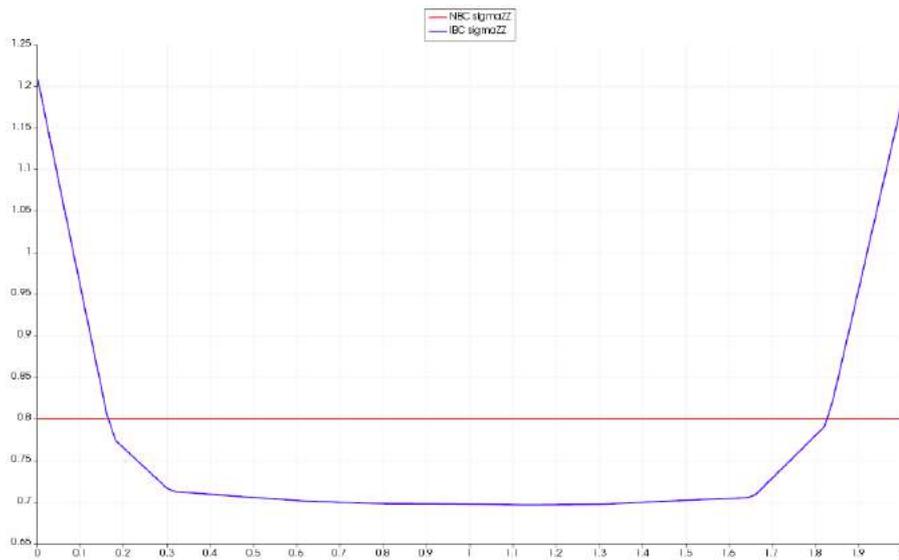


Figure 24.4 – Stress for the prism test case on Γ_I .

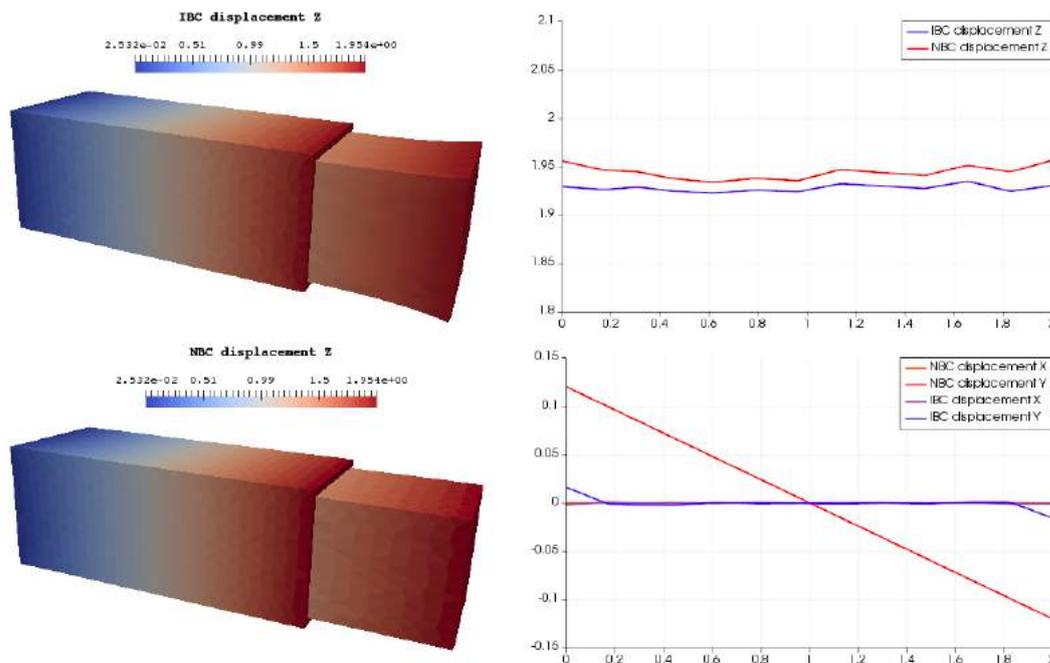


Figure 24.5 – Displacement for the prism test case on Γ_I .

Observing the behavior of the stress, Fig. 24.4 shows a constant stress for NBC, which is expected because it is actually what we are imposing as boundary condition, while we have a non-constant stress for IBC especially on the borders. This fact may be related to the fact that the stress accumulated within the body cannot be expelled deforming the shape of the boundary surface since the IBC tries to preserve the dimension of the boundary; indeed, reminding Eq. (5.8), the displacement for IBC has to be constant, while for NBC we do not have this constraint.

Fig. 24.5 reports a total view and a cross section on Γ_L of the behavior of the displacement. For the main component we do not see any appreciable difference even if we notice that NBC has developed a slightly higher value of the displacement, but the most interesting observation

can be made on the figure bottom right. As expected by the theory the other two components of the displacement in the IBC case are still constant in space, while NBC test provides a non-constant displacement on the surface of interest.

2.2 Cylinder test case

Let now consider a different geometry: the cylinder. Similar considerations as before can be done about the accuracy for the algorithm. However in this case we highlight the z-component of the force, indeed from an analytical viewpoint it should be $F_z = g |\Gamma_L| \approx 2,513$. Beside this observation, we notice in Tab. 24.2 that in the NBC test case, when we are increasing the geometric order also the accuracy of the computed force is increasing as expected.

NBC - G1		NBC - G2	
$\mathbf{F}_{NBC} = \begin{bmatrix} -1.126063 \cdot 10^{-6} \\ -6.229426 \cdot 10^{-5} \\ 2.496881 \end{bmatrix}$		$\mathbf{F}_{NBC} = \begin{bmatrix} -7.597985 \cdot 10^{-5} \\ 1.271121 \cdot 10^{-4} \\ 2.507272 \end{bmatrix}$	
IBC - EXACT	IBC - G1	IBC - G2	
$\mathbf{F}_{IBC} = \begin{bmatrix} -2.91 \cdot 10^{-5} \\ 9.304520 \cdot 10^{-5} \\ 2.512181 \end{bmatrix}$	$\mathbf{F}_{IBC} = \begin{bmatrix} 1.752510 \cdot 10^{-4} \\ 2.144581 \cdot 10^{-4} \\ 2.496223 \end{bmatrix}$	$\mathbf{F}_{IBC} = \begin{bmatrix} 5.786183 \cdot 10^{-6} \\ 4.960038 \cdot 10^{-5} \\ 2.516996 \end{bmatrix}$	

Table 24.2 – Table of results for the cylinder test case.

For what concerns the IBC test case, we have to be careful in the analysis, indeed we are imposing directly the condition on the force and if we impose the analytical value, which is the exact one, the stress is self-adjusting in order to reach the exact value of the force on that boundary, because we are imposing that condition in a strong way. In order to compare the two test cases, we have made two further simulations imposing this time the numerical value for IBC, which means that we have imposed $F_z = g |\Gamma_L|$ but with $|\Gamma_L|$ computed on the mesh. The interesting remark is that we are not imposing the exact value strongly and in this way the stress is not self-adjusted and as expected the result for $G = 1$ is not good as before, but still the computed force obtained with $G = 2$ in this case is more accurate than in the NBC simulation.

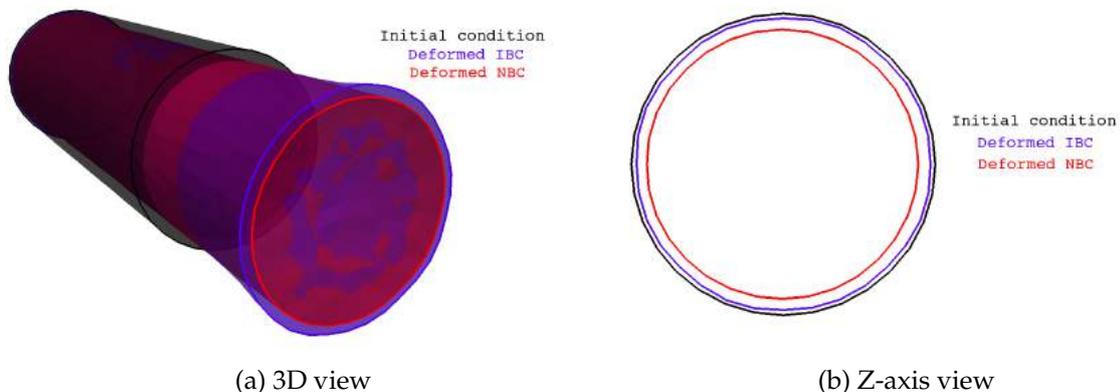


Figure 24.6 – Initial and final condition for the cylinder test case.



Figure 24.7 – Stress in the cylinder test case on Γ_L .

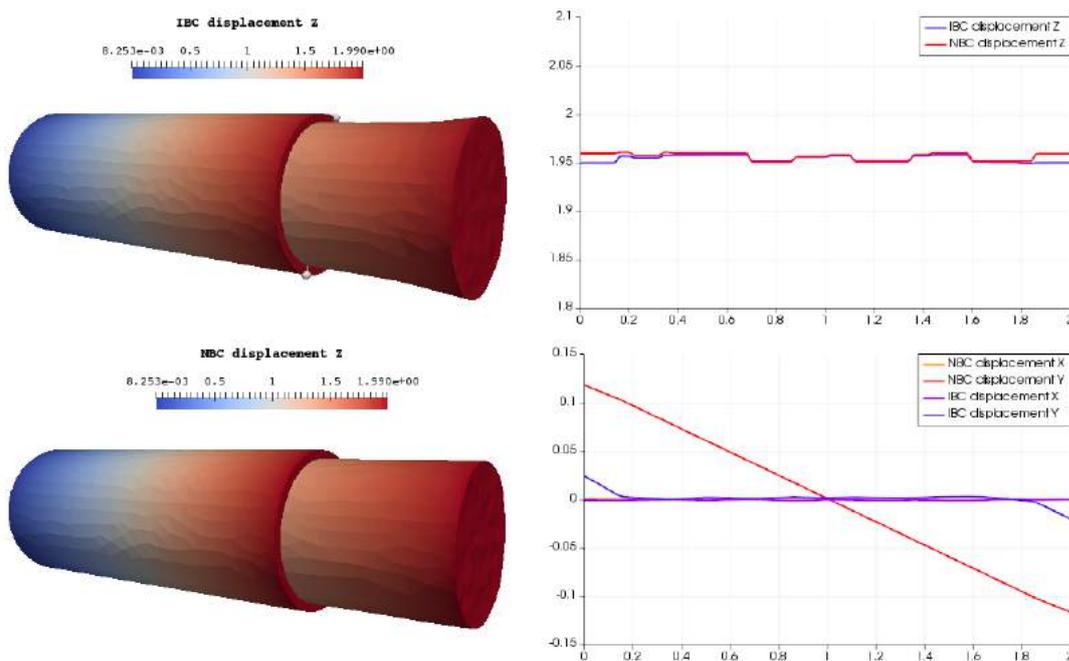


Figure 24.8 – Displacement in the cylinder test case on Γ_L .

For what concerns the qualitative displacement along the main direction and the final shape of Γ_L that is the surface of interest, same remarks as with the previous geometry are valid (see Figs. 24.6a and 24.6b). As proof of our observations made above, Fig 24.7 shows again a constant behavior of the stress in NBC test case, while IBC test presents the border effect as previously explained. The displacement, illustrated in Fig. 24.8, has the same behavior as in the prism test case, even if with this geometry we cannot say that the displacement is bigger for NBC than for IBC.

3 Conclusions

In the conclusion we have shown that imposing the integral boundary condition preserves the dimension of the surface where the force is applied, while the Neumann boundary condition let the body with less constraint. As predicted by the theory, IBC provides constant displacement and non-constant stress on that surface, meanwhile with NBC we obtain constant stress and non-constant displacement.

We envision, for instance, the use of the IBC for elasticity to couple a 3D structure - 0D rigid body. Thus, this multiscale model is connected naturally, in particular the stress of the elastic solid influences the rigid body motion described by a system of ODEs and *vice-versa*.

Part VIII

Conclusions and outlook

Conclusions and outlook

In this thesis we have presented an innovative mathematical and computational framework in the biomedical context of ophthalmology.

Starting from the current challenges in ocular diseases, in particular glaucoma, we have modeled the hemodynamics and the biomechanics of the principal tissues of the eye. The great innovation we have brought from a modeling viewpoint is the coupling of various multi-physics and multiscale models under a unified framework that we called Ocular Mathematical Virtual Simulator (OMVS). The key of success of this ambitious project is the modularity of the overall system, which provides the possibility to replace and incorporate new or more advanced models maintaining the same computational framework. These significant results have been obtained thanks to novel numerical methods we have developed, which are a good fit in such mathematical context. Specifically, the development of a Hybridizable Discontinuous Galerkin discretization approach with non-standard boundary condition and a 3D-0D operator splitting algorithm granted the user to achieve high accuracy in a flexible environment without exceeding in the computational cost or losing stability.

The limitations of our model are associated with the geometrical and mathematical approximation we had considered in the development. In particular, the geometry has been generated from a CAD image of a general patient, whereas for a more precise analysis the computational mesh should be reconstructed from patient-specific images such as the MRI. The hemodynamics in the ocular vessels (CRA, CRV, ophthalmic artery and vein, etc.) is modeled *via* a 0D model, which simplifies the physiology - however not necessarily the accuracy - and assumes the hypothesis listed in Ch. 3. The biomechanics of the ocular tissues is simulated by way of linear elastic equations, neglecting - at this first stage - nonlinear and viscoelastic effects. Nevertheless we provide this new computational environment as a first step towards more complex mathematical model where these limitations can be addressed and weakened.

In addition throughout the thesis, as endorsement of the goodness of the proposed model

despite the limitations indicated above, we have completed the VVUQ (Verification Validation and Uncertainty Quantification) process :

- **Verification.** We verified in Ch. 12 that the innovative HDG methods with integral boundary condition applied to the Darcy equations and the linear elasticity system reached the predicted order of convergence. In a similar manner, we tested the novel operator splitting method for the 3D-0D coupled problem.
- **Validation.** The whole Chapter 13 is focused on the comparison between the simulation results provided by our model and data that come from experiments - whenever available - and other mathematical models in literature.
- **Uncertainty Quantification.** In the last part of the thesis (Part VI) we illustrate a preliminary investigation towards uncertainty quantification and a sensitivity analysis based on Sobol indices for a reduced version OMVS.

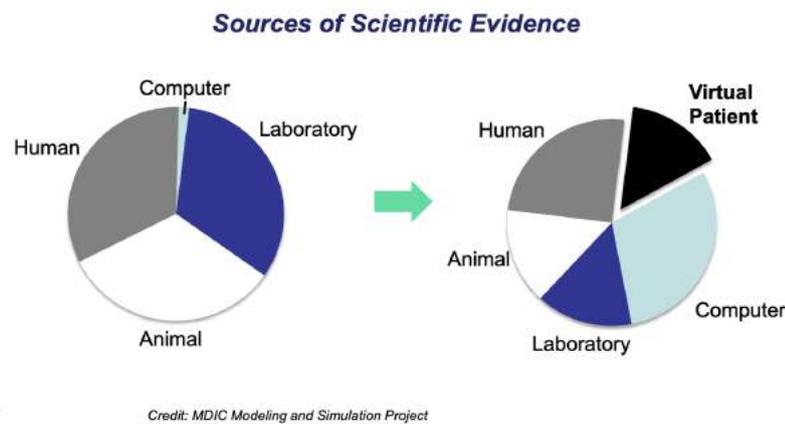


Figure 25.1 – FDA Medical Device Regulatory Evaluation. Courtesy of <https://www.fda.gov>.

In the context of computational models this process is crucial for the scientific research, especially in biology and medicine due to the fact it can decrease the number of physical tests necessary for product development or diseases understanding, thus reducing tremendously the economic cost. The U.S. Food & Drug Administration (FDA) points out that the current paradigm of medical device evaluation involves the review of scientific evidence from four models: animal data, laboratory data (or bench tests), human data (clinical trials) and data from computational modeling (in a little part). However, as shown in Fig. 25.1 that is retrieved directly from the FDA website, the future directions include more computational modeling. This fact does not necessarily imply the need of more data, rather it expresses the interest of relying less on animal and human data with a greater influence of data from virtual patients. This idea is shared also by the European Medicines Agency (EMA) in the official document explaining the plan for advancing EMA's engagement with regulatory science over the next years (*Regulatory Science to 2025* *).

Pursuing this concept, we provide a computational framework that serves as an ophthalmological virtual laboratory endowed with a user-friendly online interface (Ch. 11), suitable for clinical purposes. As next step, we envision to add to this flexible design other works that

*https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/ema-regulatory-science-2025-strategic-reflection_en.pdf

our research group is carrying out in parallel, for instance the direct integration with medical instruments such as the OCTA (Optical Coherence Tomography) [39]. Some of these data are already used, *e.g.* the retinal depth and the dimensions of the lamina cribrosa, however an expertise in OCTA imaging is currently required to the OMVS user to retrieve these data and use them as input of the model.

Moreover, we foresee some extensions in order to include

- i) the biomechanics of the vitreous humor [240];
- ii) the microcirculation in the retina and in the choroid [3];
- iii) the oxygenation of ocular tissues [244, 38];
- iv) the metabolism and electrochemistry in the optic nerve head region [208, 89, 155];
- v) the coupling with the cerebrospinal fluid vasculature (Ch. 21);
- vi) the lymphatic system [149] coupled with the model presented in Ch. 22;
- vii) the implementation of the reduce basis method to HDG in order to speed up the computation [186];
- viii) the second order convergence in time for the operator splitting method [84];
- ix) the development of the operator splitting strategy for poroelastic problems;
- x) a more detailed analysis using Sobol indices on the OMVS.

Finally we conclude renewing the goal of the project **Eye2Brain** where this thesis is embedded. The study of the eye is a first step towards the understanding of the complex cerebral mechanisms, in particular for neurodegenerative diseases. With this goal we mention the undergoing thesis of P. Ricka at the University of Strasbourg, which focuses on the mathematical modelisation and numerical simulation of cerebral blood flows and their interactions with the ocular system.

- [1] S. Abhyankar, J. Brown, E. M. Constantinescu, D. Ghosh, B. F. Smith, and H. Zhang. Petsc/ts: A modern scalable ode/dae solver library. *arXiv preprint arXiv:1806.01437*, 2018.
- [2] J. Alametsä, J. Viik, J. Alakare, A. Värri, and A. Palomäki. Ballistocardiography in sitting and horizontal positions. *Physiological Measurement*, 29(9):1071, 2008.
- [3] M. C. M. Aletti. *Mathematical modelling and simulations of the hemodynamics in the eye*. PhD thesis, Université Pierre et Marie Curie (UPMC Paris 6), 2017.
- [4] K. Ambarki, O. Baledent, G. Kongolo, R. Bouzerar, S. Fall, and M.-E. Meyer. A new lumped-parameter model of cerebrospinal hydrodynamics during the cardiac cycle in healthy volunteers. *IEEE transactions on biomedical engineering*, 54(3):483–491, 2007.
- [5] P. R. Amestoy, I. S. Duff, and J.-Y. L’Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer methods in applied mechanics and engineering*, 184(2-4):501–520, 2000.
- [6] P. R. Amestoy, I. S. Duff, and J.-Y. L’Excellent. Parallélisation de la factorisation lu de matrices creuses non-symétriques pour des architectures à mémoire distribuée. *Calculateurs Parallèles Réseaux et systèmes répartis*, 10(5):509–520, 1998.
- [7] C. Andersson, J. Åkesson, and C. Führer. *PyFMI: A Python package for simulation of coupled dynamic models with the functional mock-up interface*. Centre for Mathematical Sciences, Lund University, 2016.
- [8] D. Arnold and F. Brezzi. Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *Math. Modeling and Numer. Anal.*, 19(1):7–32, 1985.
- [9] G. Avanzolini, P. Barbini, A. Cappello, and G. Cevenini. Cadcs simulation of the closed-loop cardiovascular system. *International Journal of Biomedical Computing*, 22(1):39–49, 1988.
- [10] M. Baudin, A. Dutfoy, B. Iooss, and A.-L. Popelin. Openturns: An industrial software for uncertainty quantification in simulation. *Handbook of uncertainty quantification*, pages 2001–2038, 2017.

- [11] A. J. Bellezza, R. T. Hart, and C. F. Burgoyne. The optic nerve head as a biomechanical structure: initial finite element modeling. *Investigative ophthalmology & visual science*, 41(10):2991–3000, 2000.
- [12] C. Bertoglio, A. Caiazzo, and M. A. Fernández. Fractional-step schemes for the coupling of distributed and lumped models in hemodynamics. *SIAM Journal on Scientific Computing*, 35(3):B551–B575, 2013.
- [13] S. Bertoluzza, G. Guidoboni, R. Hild, D. Prada, C. Prud’Homme, R. Sacco, L. Sala, and M. Szopos. Hdg methods with integral boundary conditions: theory and applications.
- [14] A. Bhan, A. C. Browning, S. Shah, R. Hamilton, D. Dave, and H. S. Dua. Effect of corneal thickness on intraocular pressure measurements with the pneumotonometer, goldmann applanation tonometer, and tono-pen. *Investigative ophthalmology & visual science*, 43(5):1389–1392, 2002.
- [15] M. Biot and D. Willis. The elastic coefficients of the theory of consolidation. *J. appl. Mech*, 24:594–601, 1957.
- [16] P. Blanco and R. Feijóo. A dimensionally-heterogeneous closed-loop model for the cardiovascular system and its applications. *Medical Engineering & Physics*, 35(5):652–667, 2013.
- [17] G. Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Clermont-Ferrand 2, 2009.
- [18] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, et al. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, number 076, pages 173–184. Linköping University Electronic Press, 2012.
- [19] H. J. Bogaard, K. Abe, A. V. Noordegraaf, and N. F. Voelkel. The right ventricle under pressure: cellular and molecular mechanisms of right-heart failure in pulmonary hypertension. *Chest*, 135(3):794–804, 2009.
- [20] B. L. Boyce, J. M. Grazier, R. E. Jones, and T. D. Nguyen. Full-field deformation of bovine cornea under constrained inflation conditions. *Biomaterials*, 29(28):3896–3904, 2008.
- [21] J. D. Brandt. Corneal thickness in glaucoma screening, diagnosis, and management. *Current opinion in ophthalmology*, 15(2):85–89, 2004.
- [22] B. Brazile, B. Yang, A. G. and Po Lam, A. Voorhees, and I. A. Sigal. One beam-one vessel is not true: lamina cribrosa vessel and collagen beam networks have distinct topologies. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [23] S. Bryan, M. S.-P. Larose, N. Campbell, J. Clarke, and M. S. Tremblay. Resting blood pressure and heart rate measurement in the canadian health measures survey, cycle 1. *Health Reports*, 21(1):71, 2010.
- [24] J. Buishas, I. G. Gould, and A. A. Linninger. A computational model of cerebrospinal fluid production and reabsorption driven by starling forces. *Croatian medical journal*, 55(5):481–497, 2014.

- [25] A. C. Bunck, J. R. Kroeger, A. Juettner, A. Brentrup, B. Fiedler, G. R. Crelier, B. A. Martin, W. Heindel, D. Maintz, W. Schwindt, et al. Magnetic resonance 4d flow analysis of cerebrospinal fluid dynamics in chiari i malformation with and without syringomyelia. *European radiology*, 22(9):1860–1870, 2012.
- [26] E. Burman and M. A. Fernández. Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering*, 198(5-8):766–784, 2009.
- [27] C. Cancelli and T. Pedley. A separated-flow model for collapsible-tube oscillations. *Journal of Fluid Mechanics*, 157:375–404, 1985.
- [28] S. Čanić, C. J. Hartley, D. Rosenstrauch, J. Tambača, G. Guidoboni, and A. Mikelić. Blood flow in compliant arteries: an effective viscoelastic reduced model, numerics, and experimental validation. *Annals of Biomedical Engineering*, 34(4):575–592, 2006.
- [29] S. Čanić, J. Tambača, G. Guidoboni, A. Mikelić, C. J. Hartley, and D. Rosenstrauch. Modeling viscoelastic behavior of arterial walls and their interaction with pulsatile blood flow. *SIAM Journal on Applied Mathematics*, 67(1):164–193, 2006.
- [30] L. Carichino, G. Guidoboni, and M. Szopos. Energy-based operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows: The stokes case. *Journal of Computational Physics*, 364:235–256, 2018.
- [31] S. Cassani. *Blood circulation and aqueous humor flow in the eye: multi-scale modeling and clinical applications*. PhD thesis, 2016.
- [32] G. N. Cattermole, P. M. Leung, G. Y. Ho, P. W. Lau, C. P. Chan, S. S. Chan, B. E. Smith, C. A. Graham, and T. H. Rainer. The normal ranges of cardiovascular parameters measured using the ultrasonic cardiac output monitor. *Physiological Reports*, 5(6), 2017.
- [33] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer methods in applied mechanics and engineering*, 194(42-44):4506–4527, 2005.
- [34] P. Causin, G. Guidoboni, A. Harris, D. Prada, R. Sacco, and S. Terragni. A poroelastic model for the perfusion of the lamina cribrosa in the optic nerve head. *Mathematical biosciences*, 257:33–41, 2014.
- [35] P. Causin and R. Sacco. A discontinuous Petrov–Galerkin method with Lagrangian multipliers for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 43(1):280–302, 2005.
- [36] A. Chandekar, D. K. Mishra, S. Sharma, G. K. Saraogi, U. Gupta, and G. Gupta. 3d printing technology: A new milestone in the development of pharmaceuticals. *Current Pharmaceutical Design*, 2019.
- [37] W. Chen, X. Zhu, T. Nemoto, K.-i. Kitamura, K. Sugitani, and D. Wei. Unconstrained monitoring of long-term heart and breath rates during sleep. *Physiological Measurement*, 29(2):N1, 2008.
- [38] G. Chiaravalli. A virtual laboratory for retinal physiology: a theoretical study of retinal oxygenation in healthy and disease. 2018.

- [39] G. Chiaravalli, G. Guidoboni, R. Sacco, T. A. Ciulla, and A. Harris. A theoretical study of vascular configurations of retinal capillary plexi based on octa data. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [40] T. Chuangsuwanich, K. E. Birgersson, A. Thiery, S. G. Thakku, H. L. Leo, and M. J. Girard. Factors influencing lamina cribrosa microcapillary hemodynamics and oxygen concentrations. *Investigative ophthalmology & visual science*, 57(14):6167–6179, 2016.
- [41] M. Ciancaglini, P. Carpineto, C. Costagliola, and L. Matropasqua. Perfusion of the optic nerve head and visual field damage in glaucomatous patients. *Graefe's archive for clinical and experimental ophthalmology*, 239(8):549–555, 2001.
- [42] T. A. Ciulla, A. Harris, P. Latkany, H. C. Piper, O. Arend, H. Garzosi, and B. Martin. Ocular perfusion abnormalities in diabetes. *Acta Ophthalmologica Scandinavica*, 80(5):468–477, 2002.
- [43] B. Cockburn. The hybridizable discontinuous galerkin methods. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 2749–2775. World Scientific, 2010.
- [44] B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable galerkin method for second-order elliptic problems. *Math. Comp.*, 77(264):1887–1916, 2008.
- [45] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, and R. Sacco. A hybridizable Discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. *SIAM Journal on Scientific Computing*, 31(5):3827–3846, 2009.
- [46] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, 2009.
- [47] B. Cockburn, J. Gopalakrishnan, and F. Sayas. A projection based error analysis of hdg methods. *Math. of Comp.*, 79(271):1351–1367, 2010.
- [48] B. Cockburn, J. Guzmán, and H. Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Math. Comp.*, 78(265):1–24, 2009.
- [49] B. Cockburn and C.-W. Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
- [50] T. Colton and F. Ederer. The distribution of intraocular pressures in the general population. *Survey of ophthalmology*, 25(3):123–129, 1980.
- [51] V. P. Costa, E. S. Arcieri, and A. Harris. Blood pressure and glaucoma. *British Journal of Ophthalmology*, 93(10):1276–1282, 2009.
- [52] O. Coussy. *Poromechanics*. John Wiley & Sons, 2004.
- [53] T. Crestaux, O. Le Maître, and J.-M. Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, 2009.
- [54] F. Cyrillo and J. Greve. The effects of virtual reality on the rehabilitation of patients with knee oa: A randomized controlled clinical trial. *Annals of Physical and Rehabilitation Medicine*, 61:e129–e130, 2018.

- [55] J. E. Davies, J. Alastruey, D. P. Francis, N. Hadjiloizou, Z. I. Whinnett, C. H. Manisty, J. Aguado-Sierra, K. Willson, R. A. Foale, I. S. Malik, et al. Attenuation of wave reflection by wave entrapment creates a 'horizon effect' in the human aorta. *Hypertension*, 60(3):778–785, 2012.
- [56] J. E. Davies, N. Hadjiloizou, D. Leibovich, A. Malaweera, J. Alastruey-Armon, Z. I. Whinnett, C. H. Manisty, D. P. Francis, J. Aguado-Sierra, R. A. Foale, et al. Importance of the aortic reservoir in determining the shape of the arterial pressure waveform—the forgotten lessons of frank. *Artery Research*, 1(2):40–45, 2007.
- [57] H. Davson, G. Hollingsworth, and M. Segal. The mechanism of drainage of the cerebrospinal fluid. *Brain*, 93(4):665–678, 1970.
- [58] R. De Boer. *Theory of porous media: highlights in historical development and current state*. Springer Science & Business Media, 2012.
- [59] B. M. F. de Veubeke. Displacement and equilibrium models in the finite element method. In O. Zienkiewicz and G. Holister, editors, *Stress Analysis*, pages 145–197. John Wiley & Sons, New York, 1965.
- [60] S. Deparis, M. A. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):601–616, 2003.
- [61] E. T. Detorakis and I. G. Pallikaris. Ocular rigidity: biomechanical role, in vivo measurements and clinical significance. *Clinical & experimental ophthalmology*, 41(1):73–81, 2013.
- [62] G. T. Dorner, E. Polska, G. Garhöfer, C. Zawinka, B. Frank, and L. Schmetterer. Calculation of the diameter of the central retinal artery from noninvasive measurements in humans. *Current eye research*, 25(6):341–345, 2002.
- [63] J. C. Downs, M. D. Roberts, C. F. Burgoyne, and R. T. Hart. Multiscale finite element modeling of the lamina cribrosa microarchitecture in the eye. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4277–4280. IEEE, 2009.
- [64] N. Ehlers, T. Bramsen, and S. Sperling. Applanation tonometry and central corneal thickness. *Acta ophthalmologica*, 53(1):34–43, 1975.
- [65] A. Eklund, B. Lundkvist, L.-O. Koskinen, and J. Malm. Infusion technique can be used to distinguish between dysfunction of a hydrocephalus shunt system and a progressive dementia. *Medical and Biological Engineering and Computing*, 42(5):644–649, 2004.
- [66] S. Epstein, M. Willemet, P. J. Chowienczyk, and J. Alastruey. Reducing the number of parameters in 1d arterial blood flow modeling: less is more for patient-specific simulations. *American Journal of Physiology-Heart and Circulatory Physiology*, 309(1):H222–H234, 2015.
- [67] C. R. Ethier, M. Johnson, and J. Ruberti. Ocular biomechanics and biotransport. *Annu. Rev. Biomed. Eng.*, 6:249–273, 2004.
- [68] D. W. Evans, A. Harris, M. Garrett, H. S. Chung, and L. Kagemann. Glaucoma patients demonstrate faulty autoregulation of ocular blood flow during posture change. *British Journal of Ophthalmology*, 83(7):809–813, 1999.

- [69] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
- [70] C. A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. Hughes, and C. A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer methods in applied mechanics and engineering*, 195(41-43):5685–5706, 2006.
- [71] O. Findl, K. Strenn, M. Wolzt, R. Menapace, C. Vass, H.-G. Eichler, and L. Schmetterer. Effects of changes in intraocular pressure on human ocular haemodynamics. *Current eye research*, 16(10):1024–1029, 1997.
- [72] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular mathematics, volume 1 of modeling, simulation and applications*, 2009.
- [73] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, volume 1. Springer Science & Business Media, 2010.
- [74] J. Fouchet-Incaux, C. Grandmont, and S. Martin. Numerical stability of coupling schemes in the 3d/0d modelling of airflows and blood flows. 2014.
- [75] T. R. Friberg and J. W. Lace. A comparison of the elastic properties of human choroid and sclera. *Experimental eye research*, 47(3):429–436, 1988.
- [76] P. Fritzson, P. Aronsson, H. Lundvall, K. Nyström, A. Pop, L. Saldamli, and D. Broman. The OpenModelica modeling, simulation, and development environment. In *46th Conference on Simulation and Modelling of the Scandinavian Simulation Society (SIMS2005), Trondheim, Norway, October 13-14, 2005*, 2005.
- [77] F. Galassi, A. Sodi, F. Ucci, A. Harris, and H. S. Chung. Ocular haemodynamics in glaucoma associated with high myopia. *International ophthalmology*, 22(5):299–305, 1998.
- [78] S. Garnotel, S. Salmon, and O. Balédent. Numerical modeling of the intracranial pressure using windkessel models. *MathS In Action*, 8:9–25, 01 2017.
- [79] M. Gehlen, V. Kurtcuoglu, and M. S. Daners. Is posture-related craniospinal compliance shift caused by jugular vein collapse? a theoretical analysis. *Fluids and Barriers of the CNS*, 14(1):5, 2017.
- [80] J.-F. Gerbeau and M. Vidrascu. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):631–647, 2003.
- [81] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.
- [82] L. Giovannardi, O. T. Inan, R. M. Wiard, M. Etemadi, and G. T. Kovacs. Ballistocardiography - a method worth revisiting. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4279–4282. IEEE, 2011.
- [83] M. J. Girard, J.-K. F. Suh, M. Bottlang, C. F. Burgoyne, and J. C. Downs. Biomechanical changes in the sclera of monkey eyes exposed to chronic iop elevations. *Investigative ophthalmology & visual science*, 52(8):5656–5669, 2011.

- [84] R. Glowinski. Finite element methods for incompressible viscous flow. *Handbook of numerical analysis*, 9:3–1176, 2003.
- [85] R. Glowinski, E. Dean, G. Guidoboni, L. Juarez, and T.-W. Pan. Applications of operator-splitting methods to the direct numerical simulation of particulate and free-surface flows and to the numerical solution of the two-dimensional elliptic monge-ampere equation. *Japan journal of industrial and applied mathematics*, 25(1):1, 2008.
- [86] J. B. Grotberg and O. E. Jensen. Biofluid mechanics in flexible tubes. *Annu. Rev. Fluid Mech.*, 36:121–147, 2004.
- [87] A. Grove. *Physics and Technology of Semiconductor Devices*. Wiley, 1967.
- [88] R. Grytz, G. Meschke, and J. B. Jonas. The collagen fibril architecture in the lamina cribrosa and peripapillary sclera predicted by a computational remodeling approach. *Biomechanics and modeling in mechanobiology*, 10(3):371–382, 2011.
- [89] G. Guidoboni, G. Bonifazi, R. Sacco, A. Layton, S. D. Olson, M. C. B. Hallare, B. A. Siesky, C. Bruttini, A. C. Verticchio Vercellin, and A. Harris. Electrochemical characterization of ciliary epithelium physiology: a theoretical approach. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [90] G. Guidoboni, A. Harris, L. Carichino, Y. Arieli, and B. A. Siesky. Effect of intraocular pressure on the hemodynamics of the central retinal artery: a mathematical model. *Mathematical Biosciences & Engineering*, 11(3):523–546, 2014.
- [91] G. Guidoboni, A. Harris, S. Cassani, J. Arciero, B. Siesky, A. Amireskandari, L. Tobe, P. Egan, I. Januleviciene, and J. Park. Intraocular pressure, blood pressure, and retinal blood flow autoregulation: a mathematical model to clarify their relationship and clinical relevance. *Investigative ophthalmology & visual science*, 55(7):4105–4118, 2014.
- [92] G. Guidoboni, A. Harris, and R. Sacco. *Mathematical Modeling of Ocular Fluid Dynamics: From Theory to Clinical Applications*. Modeling and Simulation in Science, Engineering, and Technology. Springer-Birkhauser (New York), July, 2019.
- [93] G. Guidoboni, C. Prud’homme, L. Sala, and M. Szopos. Unconditionally stable operator splitting method for a multiscale application in ophthalmology. In *44e Congrès National d’Analyse Numérique*, 2018.
- [94] G. Guidoboni, L. Sala, M. Enayati, R. Sacco, M. Szopos, J. M. Keller, M. Popescu, L. Despins, V. Huxley, and M. Skubic. Cardiovascular function and ballistocardiogram: a relationship interpreted via mathematical modeling. *IEEE Transactions on Biomedical Engineering*, 2019.
- [95] J. Guinane. An equivalent circuit analysis of cerebrospinal fluid hydrodynamics. *American Journal of Physiology-Legacy Content*, 223(2):425–430, 1972.
- [96] S. Gupta, M. Soellinger, D. M. Grzybowski, P. Boesiger, J. Biddiscombe, D. Poulikakos, and V. Kurtcuoglu. Cerebrospinal fluid dynamics in the human cranial subarachnoid space: an overlooked mediator of cerebral disease. i. computational model. *Journal of the Royal Society Interface*, 7(49):1195–1204, 2010.
- [97] E. A. Hadzri, K. Osman, M. R. A. Kadir, and A. A. Aziz. Computational investigation on csf flow analysis in the third ventricle and aqueduct of sylvius. *IIUM Engineering Journal*, 12(3), 2011.

- [98] S. Hales. Statistical essays: concerning haemastaticks; or, an account of some hydraulick and hydrostatical experiments made on the blood and blood-vessels of animals. *London: W. Innys and R. Manby, 1733, 1733.*
- [99] A. Harris, R. B. Dinn, L. Kagemann, and E. Rechtman. A review of methods for human retinal oximetry. *Ophthalmic Surgery, Lasers and Imaging Retina*, 34(2):152–164, 2003.
- [100] A. Harris, M. Harris, J. Biller, H. Garzozi, D. Zarfty, T. A. Ciulla, and B. Martin. Aging affects the retrobulbar circulation differently in women and men. *Archives of Ophthalmology*, 118(8):1076–1080, 2000.
- [101] A. Harris, K. Joos, M. Kay, D. Evans, R. Shetty, W. E. Sponsel, and B. Martin. Acute iop elevation with scleral suction: effects on retrobulbar haemodynamics. *British journal of ophthalmology*, 80(12):1055–1059, 1996.
- [102] A. Harris, R. Sergott, G. Spaeth, J. Katz, J. Shoemaker, and B. Martin. Color doppler analysis of ocular vessel blood velocity in normal-tension glaucoma. *American journal of ophthalmology*, 118(5):642–649, 1994.
- [103] D. Harris. The inheritance of glaucoma*: A pedigree of familial glaucoma. *American journal of ophthalmology*, 60(1):91–95, 1965.
- [104] Z. He, C. T. Nguyen, J. A. Armitage, A. J. Vingrys, and B. V. Bui. Blood pressure modifies retinal susceptibility to intraocular pressure elevation. *PLoS One*, 7(2):e31104, 2012.
- [105] M. Helfand, V. Christensen, and J. Anderson. Technology assessment: Early Sense for monitoring vital signs in hospitalized patients. 2016.
- [106] L. W. Herndon, J. S. Weizer, and S. S. Stinnett. Central corneal thickness as a risk factor for advanced glaucoma damage. *Archives of ophthalmology*, 122(1):17–21, 2004.
- [107] M. Hirschvogel, M. Bassilious, L. Jagschies, S. M. Wildhirt, and M. W. Gee. A monolithic 3d-0d coupled closed-loop model of the heart and the vascular system: Experiment-based parameter estimation for patient-specific cardiac mechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 33(8):e2842, 2017.
- [108] L. Howden, D. Giddings, and H. Power. Three-dimensional cerebrospinal fluid flow within the human central nervous system. *Discrete and continuous dynamical systems*, 2011.
- [109] Y. Hsu, H. M. Hettiarachchi, D. C. Zhu, and A. A. Linninger. The frequency and magnitude of cerebrospinal fluid pulsations influence intrathecal drug distribution: key factors for interpatient variability. *Anesthesia & Analgesia*, 115(2):386–394, 2012.
- [110] M. F. Huffaker, M. Carchia, B. U. Harris, W. C. Kethman, T. E. Murphy, C. C. Sakarovitch, F. Qin, and D. N. Cornfield. Passive nocturnal physiologic monitoring enables early detection of exacerbations in asthmatic children: A proof of concept study. *American Journal of Respiratory and Critical Care Medicine*, (ja), 2018.
- [111] F. H. Huhnke, J. D. Russo, P. Wyszomolek, J. Diemer, N. Dhanovici, J. P. Spatz, and F. Serwane. The tissue within a retina organoid shows a yield stress, the amount of mechanical stress necessary to induce irreversible deformations, which regulates the retina’s mechanical integrity. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.

- [112] O. Inan, M. Etemadi, R. Wiard, L. Giovangrandi, and G. Kovacs. Robust ballistocardiogram acquisition for home monitoring. *Physiological Measurement*, 30(2):169, 2009.
- [113] O. T. Inan, P.-F. Migeotte, K.-S. Park, M. Etemadi, K. Tavakolian, R. Casanella, J. M. Zanetti, J. Tank, I. Funtova, G. K. Prisk, et al. Ballistocardiography and seismocardiography: a review of recent advances. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1414–1427, 2015.
- [114] T. Ishigami and T. Homma. An importance quantification technique in uncertainty analysis for computer models. In [1990] *Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, pages 398–403. IEEE, 1990.
- [115] S. G. Izard, J. A. Juanes, F. J. G. Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto. Virtual reality as an educational and training tool for medicine. *Journal of medical systems*, 42(3):50, 2018.
- [116] J. D. Jackson. *Classical Electrodynamics Third Edition*. Wiley, third edition, 1998.
- [117] E. E. Jacobson, D. F. Fletcher, M. K. Morgan, and I. H. Johnston. Fluid dynamics of the cerebral aqueduct. *Pediatric neurosurgery*, 24(5):229–236, 1996.
- [118] K. Jain, G. Ringstad, P.-K. Eide, and K.-A. Mardal. Direct numerical simulation of transitional hydrodynamics of the cerebrospinal fluid in chiari i malformation—the role of cranio-vertebral junction. *International journal for numerical methods in biomedical engineering*, 2016.
- [119] A. Q. Javaid, H. Ashouri, S. Tridandapani, and O. T. Inan. Elucidating the hemodynamic origin of ballistocardiographic forces: Toward improved monitoring of cardiovascular health at home. *IEEE Journal of Translational Engineering in Health and Medicine*, 4:1–8, 2016.
- [120] N. Johnson, S. Kotz, and N. Balakrishnan. Continuous volume1, 1994.
- [121] I. Jones, M. Warner, and J. Stevens. Mathematical modelling of the elastic properties of retina: a determination of young’s modulus. *Eye*, 6(6):556, 1992.
- [122] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.
- [123] Y. Katz, R. Karasik, and Z. Shinar. Contact-free piezo electric sensor used for real-time analysis of inter beat interval series. In *Computing in Cardiology Conference (CinC), 2016*, pages 769–772. IEEE, 2016.
- [124] R. C. Kerckhoffs, M. L. Neal, Q. Gu, J. B. Bassingthwaighte, J. H. Omens, and A. D. McCulloch. Coupling of a 3d finite element model of cardiac ventricular mechanics to lumped systems models of the systemic and pulmonic circulation. *Annals of Biomedical Engineering*, 35(1):1–18, 2007.
- [125] J. Kerr, P. Nelson, and C. O’Brien. A comparison of ocular blood flow in untreated primary open-angle glaucoma and ocular hypertension. *American journal of ophthalmology*, 126(1):42–51, 1998.
- [126] C.-S. Kim, S. L. Ober, M. S. McMurtry, B. A. Finegan, O. T. Inan, R. Mukkamala, and J.-O. Hahn. Ballistocardiogram: Mechanism and potential for unobtrusive cardiovascular health monitoring. *Scientific Reports*, 6:31297, 2016.

- [127] D.-J. Kim, Z. Czosnyka, N. Keong, D. K. Radolovich, P. Smielewski, M. P. Sutcliffe, J. D. Pickard, and M. Czosnyka. Index of cerebrospinal compensatory reserve in hydrocephalus. *Neurosurgery*, 64(3):494–502, 2009.
- [128] R. M. Kirby, S. J. Sherwin, and B. Cockburn. To CG or to HDG: A comparative study. *Journal of Scientific Computing*, 51(1):183–212, 2012.
- [129] R. Klabunde. *Cardiovascular physiology concepts*. Lippincott Williams & Wilkins, 2011.
- [130] O. Kosoko-Lasaki, G. Gong, G. Haynatzki, and M. R. Wilson. Race, ethnicity and prevalence of primary open-angle glaucoma. *Journal of the National Medical Association*, 98(10):1626, 2006.
- [131] A. Kotecha, E. White, J. Shewry, and D. Garway-Heath. The relative effects of corneal thickness and age on goldmann applanation tonometry and dynamic contour tonometry. *British Journal of Ophthalmology*, 89(12):1572–1575, 2005.
- [132] V. Kurtcuoglu. *Biomechanics of the Brain*, chapter 8: Computational fluid dynamics for the assessment of cerebrospinal fluid flow and its coupling with cerebral blood flow. Springer New York, 2011.
- [133] V. Kurtcuoglu, M. Soellinger, P. Summers, K. Boomsma, D. Poulikakos, P. Boesiger, and Y. Ventikos. Computational investigation of subject-specific cerebrospinal fluid flow in the third ventricle and aqueduct of sylvius. *Journal of biomechanics*, 40(6):1235–1245, 2007.
- [134] W. D. Lakin, S. A. Stevens, B. I. Tranmer, and P. L. Penar. A whole-body mathematical model for intracranial pressure dynamics. *Journal of Mathematical Biology*, 46(4):347–383, 2003.
- [135] R. K. Lambert, T. A. Wilson, R. E. Hyatt, and J. R. Rodarte. A computational model for expiratory flow. *Journal of Applied Physiology*, 52(1):44–56, 1982.
- [136] G. Lamé. *Leçons sur la théorie mathématique de l'élasticité des corps solides par G. Lamé*. Gauthier-Villars, 1866.
- [137] G. Lazaridis, J. Mohamed-Noriega, and D. F. Garway-Heath. Bilateral disc haemorrhages in the united kingdom glaucoma treatment study (ukgts): a probabilistic approach to explore a possible systemic pathophysiological mechanism. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [138] P. Y. Lee, Z. He, V. H. Y. Wong, J. G. Crowston, and B. V. Bui. The effect of ageing on the recovery of retinal function and structure following intraocular pressure elevation in mice. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [139] M. C. Leske, F. Ederer, and M. Podgor. Estimating incidence from age-specific prevalence in glaucoma. *American journal of epidemiology*, 113(5):606–613, 1981.
- [140] F. Liang and H. Liu. A closed-loop lumped parameter computational model for human cardiovascular system. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 48(4):484–493, 2005.
- [141] E. Lifesciences. Normal hemodynamic parameters and laboratory values. Retrieved on April, 9, 2014.

- [142] E. Limpert, W. A. Stahel, and M. Abbt. Log-normal distributions across the sciences: keys and clues: on the charms of statistics, and how mechanical models resembling gambling machines offer a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability—normal or log-normal: that is the question. *BioScience*, 51(5):341–352, 2001.
- [143] A. Linniger, C. Tsakiris, D. Zhu, M. Xenos, P. Roycewicz, Z. Danziger, and R. Penn. Pulsatile cerebrospinal fluid flow in the normal and abnormal intracranial system. *IEEE Transaction on Biomedical Engineering*, 2005.
- [144] A. A. Linninger, K. Tangen, C.-Y. Hsu, and D. Frim. Cerebrospinal fluid mechanics and its coupling to cerebrovascular dynamics. *Annual Review of Fluid Mechanics*, 48:219–257, 2016.
- [145] A. A. Linninger, M. Xenos, B. Sweetman, S. Ponshe, X. Guo, and R. Penn. A mathematical model of blood, cerebrospinal fluid and brain dynamics. *Journal of mathematical biology*, 59(6):729–759, 2009.
- [146] A. Maceira, S. Prasad, M. Khan, and D. Pennell. Normalized left ventricular systolic and diastolic function by steady state free precession cardiovascular magnetic resonance. *Journal of Cardiovascular Magnetic Resonance*, 8(3):417–426, 2006.
- [147] A. M. Maceira, S. K. Prasad, M. Khan, and D. J. Pennell. Reference right ventricular systolic and diastolic function normalized to age, gender and body surface area from steady-state free precession cardiovascular magnetic resonance. *European Heart Journal*, 27(23):2879–2888, 2006.
- [148] Z. Mao, A. Miki, S. Mei, Y. Dong, K. Maruyama, R. Kawasaki, S. Usui, K. Matsushita, K. Nishida, and K. Chan. Automated 3d lamina cribrosa segmentation in optical coherence tomography volumetric scans. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [149] N. Marazzi. An engineering approach to the multiscale modeling of the lymphatic system. 2018.
- [150] B. Marek, A. Harris, P. Kanakamedala, E. Lee, A. Amireskandari, L. Carichino, G. Guidoboni, L. A. Tobe, and B. Siesky. Cerebrospinal fluid pressure and glaucoma: regulation of trans-lamina cribrosa pressure. *British Journal of Ophthalmology*, 98(6):721–725, 2014.
- [151] B. Markert. *Porous media viscoelasticity with application to polymeric foams*. 2005.
- [152] P. Markowich, C. Ringhofer, and C. Schmeiser. *Semiconductor Equations*. Springer-Verlag, 1990.
- [153] A. Marmarou, K. Shulman, and J. Lamorgese. Compartmental analysis of compliance and outflow resistance of the cerebrospinal fluid system. *Journal of neurosurgery*, 43(5):523–534, 1975.
- [154] A. Marmarou, K. Shulman, and R. M. Rosende. A nonlinear analysis of the cerebrospinal fluid system and intracranial pressure dynamics. *Journal of neurosurgery*, 48(3):332–344, 1978.

- [155] S. Mathew, A. Harris, A. C. Verticchio Vercellin, B. Wirostko, B. Martin, and G. Guidoboni. Relationship between episcleral venous pressure, intraocular pressure and trabecular meshwork resistance: a mathematical model. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [156] T. Matsuyama, Y. Kinugasa, Y. Nakajima, and K. Kojima. Robotic-assisted surgery-current state of the art and future perspective colorectal cancer. *Gan to kagaku ryoho. Cancer & chemotherapy*, 45(12):1696–1700, 2018.
- [157] D. McDonald. Blood flow in arteries. 1974. *Edward Arnold, London*, pages 92–95.
- [158] M. E. Moghadam, I. E. Vignon-Clementel, R. Figliola, A. L. Marsden, M. of Congenital Hearts Alliance (MOCHA) Investigators, et al. A modular numerical method for implicit 0d/3d coupling in cardiovascular finite element simulations. *Journal of Computational Physics*, 244:63–79, 2013.
- [159] W. Morgan, J. Khoo, A. Kermode, C. Lind, M. Hazelton, K. Parsons, and D. Yu. Utilization of retinal vein photoplethysmography to measure intracranial pressure. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [160] W. H. Morgan, B. C. Chauhan, D.-Y. Yu, S. J. Cringle, V. A. Alder, and P. H. House. Optic disc movement with variations in intraocular and cerebrospinal fluid pressure. *Investigative ophthalmology & visual science*, 43(10):3236–3242, 2002.
- [161] W. H. Morgan, D.-Y. Yu, V. A. Alder, S. J. Cringle, R. L. Cooper, P. H. House, and I. J. Constable. The correlation between cerebrospinal fluid pressure and retrolaminar tissue pressure. *Investigative ophthalmology & visual science*, 39(8):1419–1428, 1998.
- [162] W. H. Morgan, D.-Y. Yu, R. L. Cooper, V. A. Alder, S. J. Cringle, and I. J. Constable. Retinal artery and vein pressures in the dog and their relationship to aortic, intraocular, and cerebrospinal fluid pressures. *Microvascular research*, 53(3):211–221, 1997.
- [163] R. Muller, T. Kamins, and M. Chan. *Device Electronics for Integrated Circuits*. Wiley, 2003.
- [164] K. M. Myers, B. Coudrillier, B. L. Boyce, and T. D. Nguyen. The inflation response of the posterior bovine sclera. *Acta biomaterialia*, 6(11):4327–4335, 2010.
- [165] N. Nagarajan, A. Dupret-Bories, E. Karabulut, P. Zorlutuna, and N. E. Vrana. Enabling personalized implant and controllable biosystem development through 3d printing. *Biotechnology advances*, 36(2):521–533, 2018.
- [166] E. S. Nelson, L. Mulugeta, A. Feola, J. Raykin, J. G. Myers, B. C. Samuels, and C. R. Ethier. The impact of ocular hemodynamics and intracranial pressure on intraocular pressure during acute gravitational changes. *Journal of Applied Physiology*, 123(2):352–363, 2017.
- [167] T. Newson and A. El-Sheikh. Mathematical modeling of the biomechanics of the lamina cribrosa under elevated intraocular pressures. *Journal of biomechanical engineering*, 128(4):496–504, 2006.
- [168] S. Ngo, A. Harris, B. A. Siesky, A. Schroeder, G. Eckert, and S. Holland. Blood pressure, ocular perfusion pressure, and body mass index in glaucoma patients. *European journal of ophthalmology*, 23(5):664–669, 2013.

- [169] T. D. Nguyen, D. Midgett, B. Liu, and H. A. Quigley. Analysis for the effects of glaucoma and optic nerve damage on the pressure-induced strains in the human lamina cribrosa. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [170] F. Nobile and C. Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized robin conditions. *SIAM Journal on Scientific Computing*, 30(2):731–763, 2008.
- [171] A. Noordergraaf and C. E. Heynekamp. Genesis of displacement of the human longitudinal ballistocardiogram from the changing blood distribution. *American Journal of Cardiology*, 2(6):748–756, 1958.
- [172] A. Noordergraaf, P. D. Verdouw, and H. B. Boom. The use of an analog computer in a circulation model. *Progress in Cardiovascular Diseases*, 5(5):419–439, 1963.
- [173] A. Noordergraaf, P. D. Verdouw, and H. B. Boom. The use of an analog computer in a circulation model. *Progress in Cardiovascular Diseases*, 5(5):419–439, 1963.
- [174] R. E. Norman, J. G. Flanagan, I. A. Sigal, S. M. Rausch, I. Tertinegg, and C. R. Ethier. Finite element modeling of the human sclera: influence on optic nerve head biomechanics and connections with glaucoma. *Experimental eye research*, 93(1):4–12, 2011.
- [175] C. J. O’Brien, R. Murphy, M. Irnaten, and A. Hopkins. The role of lamina cribrosa tissue stiffness as a fundamental biomechanical driver of pathological glaucomatous cupping. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [176] J. B. Olansen, J. Clark, D. Khoury, F. Ghorbel, and A. Bidani. A closed-loop model of the canine cardiovascular system that includes ventricular interaction. *Computers and Biomedical Research*, 33(4):260–295, 2000.
- [177] J. Paalasmaa, H. Toivonen, M. Partinen, et al. Adaptive heartbeat modeling for beat-to-beat heart rate measurement in ballistocardiograms. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1945–1952, 2015.
- [178] J. Pardaens, K. Van de Woestijne, and J. Clement. Simulation of regional lung emptying during slow and forced expirations. *Journal of applied physiology*, 39(2):191–198, 1975.
- [179] K. H. Parker. An introduction to wave intensity analysis. *Medical & Biological Engineering & Computing*, 47(2):175, 2009.
- [180] L. R. Pasquale, S. J. Loomis, R. N. Weinreb, J. H. Kang, B. L. Yaspan, J. C. Bailey, D. Gaasterland, T. Gaasterland, R. K. Lee, W. K. Scott, et al. Estrogen pathway polymorphisms in relation to primary open angle glaucoma: an analysis accounting for gender from the united states. *Molecular vision*, 19:1471, 2013.
- [181] T. J. Pedley, B. S. Brook, and R. S. Seymour. Blood pressure and flow rate in the giraffe jugular vein. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 351(1342):855–866, 1996.
- [182] E. Pinheiro, O. Postolache, and P. Girão. Theory and developments in an unobtrusive cardiovascular system representation: ballistocardiography. *The Open Biomedical Engineering Journal*, 4:201, 2010.

- [183] P. M. Pinsky, D. van der Heide, and D. Chernyak. Computational modeling of mechanical anisotropy in the cornea and sclera. *Journal of Cataract & Refractive Surgery*, 31(1):136–145, 2005.
- [184] D. Prada. *A hybridizable discontinuous Galerkin method for nonlinear porous media viscoelasticity with applications in ophthalmology*. PhD thesis, 2016.
- [185] C. Prud’Homme, V. Chabannes, V. Doyeux, M. Ismail, A. Samake, and G. Pena. Feel++: A computational framework for galerkin methods and advanced numerical methods. In *ESAIM: Proceedings*, volume 38, pages 429–455. EDP Sciences, 2012.
- [186] C. Prud’Homme, C. Daversin, C. Trophime, and S. Veys. Reduced basis methods and high performance computing. applications to non-linear multi-physics problems. 2014.
- [187] O. S. Punjabi, C. Kniestedt, R. L. Stamper, and S. C. Lin. Dynamic contour tonometry: principle and use. *Clinical & experimental ophthalmology*, 34(9):837–840, 2006.
- [188] W. Qiu, J. Shen, and K. Shi. An hdg method for linear elasticity with strong symmetric stresses. *Mathematics of Computation*, 87(309):69–93, 2018.
- [189] A. Quaini and A. Quarteroni. A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method. *Mathematical models and methods in applied sciences*, 17(06):957–983, 2007.
- [190] A. Quarteroni and L. Formaggia. Mathematical modelling and numerical simulation of the cardiovascular system. *Handbook of numerical analysis*, 12:3–127, 2004.
- [191] A. Quarteroni, S. Ragni, and A. Veneziani. Coupling between lumped and distributed models for blood flow problems. *Computing and Visualization in Science*, 4(2):111–124, 2001.
- [192] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. *Computing and Visualization in Science*, 2(4):163–197, 2000.
- [193] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Lecture Notes in Mathematics. Springer-Verlag, 1994.
- [194] A. Quarteroni and A. Veneziani. Analysis of a geometrical multiscale model based on the coupling of ode and pde for blood flow simulations. *Multiscale Modeling & Simulation*, 1(2):173–195, 2003.
- [195] A. Quarteroni, A. Veneziani, and C. Vergara. Geometric multiscale modeling of the cardiovascular system, between theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 302:193 – 252, 2016.
- [196] A. Ragauskas, L. Bartusis, I. Piper, R. Zakelis, V. Matijosaitis, K. Petrikonis, and D. Rastenyte. Improved diagnostic value of a tcd-based non-invasive icp measurement method compared with the sonographic onsd method for detecting elevated intracranial pressure. *Neurological research*, 36(7):607–614, 2014.
- [197] A. Ragauskas, V. Matijosaitis, R. Zakelis, K. Petrikonis, D. Rastenyte, I. Piper, and G. Daubaris. Clinical assessment of noninvasive intracranial pressure absolute value measurement method. *Neurology*, 78(21):1684–1691, 2012.

- [198] M. Rantz, K. Lane, L. J. Phillips, L. A. Despina, C. Galambos, G. L. Alexander, R. J. Koopman, L. Hicks, M. Skubic, and S. J. Miller. Enhanced registered nurse care coordination with sensor technology: Impact on length of stay and cost in aging in place housing. *Nursing Outlook*, 63(6):650–655, 2015.
- [199] M. J. Rantz, M. Skubic, M. Popescu, C. Galambos, R. J. Koopman, G. L. Alexander, L. J. Phillips, K. Musterman, J. Back, and S. J. Miller. A new paradigm of technology-enabled ‘vital signs’ for early detection of health change for older adults. *Gerontology*, 61(3):281–290, 2015.
- [200] S. V. Rao, K. Moutzouris, and M. Ebrahimzadeh. Nonlinear frequency conversion in semiconductor optical waveguides using birefringent, modal and quasi-phase-matching techniques. *Journal of Optics A: Pure and Applied Optics*, 6(6):569, 2004.
- [201] R. Rate. Emfit sensor technology. *Journal of Medical Systems*, 31:69–77, 2007.
- [202] F. Recrosi, R. Repetto, A. Tatone, and G. Guidoboni. Thermodynamic derivation of a non-linear poroelastic model describing hemodynamicsmechanics interplay in the lamina cribrosa. *Journal for Modeling in Ophthalmology*, 2(2):80–85, 2018.
- [203] R. Ren, J. B. Jonas, G. Tian, Y. Zhen, K. Ma, S. Li, H. Wang, B. Li, X. Zhang, and N. Wang. Cerebrospinal fluid pressure in glaucoma: a prospective study. *Ophthalmology*, 117(2):259–266, 2010.
- [204] A. Ribes and C. Caremoli. Salome platform component model for numerical simulation. In *31st Annual International Computer Software and Applications Conference (COMP-SAC 2007)*, volume 2, pages 553–564. IEEE, 2007.
- [205] L. Rosales, B. Y. Su, M. Skubic, and K. Ho. Heart rate monitoring using hydraulic bed sensor ballistocardiogram. *Journal of Ambient Intelligence and Smart Environments*, 9(2):193–207, 2017.
- [206] G. Rutkowska, V. Haughton, S. Linge, and K.-A. Mardal. Patient-specific 3d simulation of cyclic csf flow at the craniocervical region. *American Journal of Neuroradiology*, 33(9):1756–1762, 2012.
- [207] R. Sacco, G. Guidoboni, and A. G. Mauri. *A comprehensive physically-based approach to modeling in bioengineering and life sciences*.
- [208] R. Sacco, G. Guidoboni, A. G. Mauri, B. A. Siesky, and A. Harris. A theoretical investigation of the role of arachidonic acid in astrocyte vasoactive agent production. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [209] L. Sala, C. Prud’homme, G. Guidoboni, and M. Szopos. Ocular mathematical virtual simulator: A hemodynamical and biomechanical study towards clinical applications. *Journal of Coupled Systems and Multiscale Dynamics*, 6(3):241–247, 2018.
- [210] L. Sala, C. Prud’homme, G. Guidoboni, M. Szopos, B. A. Siesky, and A. Harris. Analysis of iop and csf alterations on ocular biomechanics and lamina cribrosa hemodynamics. *Investigative Ophthalmology & Visual Science*, 59(9):4475–4475, 2018.
- [211] L. Sala, C. Prud’Homme, D. Prada, F. Salerni, C. Trophime, V. Chabannes, M. Szopos, R. Repetto, S. Bertoluzza, R. Sacco, et al. Patient-specific virtual simulator of tissue perfusion in the lamina cribrosa. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2017.

- [212] L. Sala, R. Sacco, and G. Guidoboni. Multiscale nature of ocular physiology. *Journal for Modeling in Ophthalmology*, 2(1):12–18, 2018.
- [213] F. Salerni, R. Repetto, A. Harris, P. Pinsky, C. Prud'homme, M. Szopos, and G. Guidoboni. Mathematical modeling of ocular and cerebral hemo-fluid dynamics: application to viip. *Journal for Modeling in Ophthalmology*, 2(2):64–68, 2018.
- [214] A. Samsudin, N. Isaacs, M.-L. S. Tai, N. Ramli, Z. Mimiwati, and M. M. Choo. Ocular perfusion pressure and ophthalmic artery flow in patients with normal tension glaucoma. *BMC ophthalmology*, 16(1):39, 2016.
- [215] O. C. SAS. Open cascade technology, 3d modeling & numerical simulation, 2008.
- [216] J. Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1):41–52, 1997.
- [217] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer-Verlag, 1984.
- [218] H. D. Sesso, M. J. Stampfer, B. Rosner, C. H. Hennekens, J. M. Gaziano, J. E. Manson, and R. J. Glynn. Systolic and diastolic blood pressure, pulse pressure, and mean arterial pressure as predictors of cardiovascular disease risk in men. *Hypertension*, 36(5):801–807, 2000.
- [219] A. H. Shapiro. Steady flow in collapsible tubes. *Journal of Biomechanical Engineering*, 99(3):126–147, 1977.
- [220] K. Shapiro and A. Marmarou. Clinical applications of the pressure-volume index in treatment of pediatric head injuries. *Journal of neurosurgery*, 56(6):819–825, 1982.
- [221] J. Shin, B. Choi, Y. Lim, D. Jeong, and K. Park. Automatic ballistocardiogram (BCG) beat detection using a template matching approach. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1144–1146. IEEE, 2008.
- [222] R. E. Showalter. Diffusion in poro-elastic media. *Journal of mathematical analysis and applications*, 251(1):310–340, 2000.
- [223] L. Siaudvytyte, I. Januleviciene, A. Ragauskas, L. Bartusis, B. Siesky, and A. Harris. Update in intracranial pressure evaluation methods and translaminar pressure gradient role in glaucoma. *Acta ophthalmologica*, 93(1):9–15, 2015.
- [224] B. Siesky, A. Harris, L. Racette, R. Abassi, K. Chandrasekhar, L. A. Tobe, J. Behzadi, G. Eckert, A. Amireskandari, and M. Muchnik. Differences in ocular blood flow in glaucoma between patients of african and european descent. *Journal of glaucoma*, 24(2):117, 2015.
- [225] I. A. Sigal and C. R. Ethier. Biomechanics of the optic nerve head. *Experimental eye research*, 88(4):799–807, 2009.
- [226] I. A. Sigal, J. G. Flanagan, I. Tertinegg, and C. R. Ethier. Finite element modeling of optic nerve head biomechanics. *Investigative ophthalmology & visual science*, 45(12):4378–4387, 2004.

- [227] B. W. Smith, J. G. Chase, R. I. Nokes, G. M. Shaw, and G. Wake. Minimal haemodynamic system model including ventricular interaction and valve dynamics. *Medical Engineering & Physics*, 26(2):131–139, 2004.
- [228] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.
- [229] I. Starr and A. Noordergraaf. *Ballistocardiography in cardiovascular research: Physical aspects of the circulation in health and disease*. Lippincott, 1967.
- [230] I. Starr and F. C. Wood. Twenty-year studies with the ballistocardiograph: the relation between the amplitude of the first record of healthy adults and eventual mortality and morbidity from heart disease. *Circulation*, 23(5):714–732, 1961.
- [231] K.-H. Støverud, H. P. Langtangen, G. A. Ringstad, P. K. Eide, and K.-A. Mardal. Computational investigation of cerebrospinal fluid dynamics in the posterior cranial fossa and cervical subarachnoid space in patients with chiari i malformation. *PloS one*, 11(10):e0162938, 2016.
- [232] B. Y. Su, M. Enayati, K. Ho, M. Skubic, L. Despins, J. M. Keller, M. Popescu, G. Guidoboni, and M. Rantz. Monitoring the relative blood pressure using a hydraulic bed sensor system. *IEEE Transactions on Biomedical Engineering*, 2018.
- [233] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979, 2008.
- [234] W. Suh, C. Kee, N. S. Group, and K. G. Society. The distribution of intraocular pressure in urban and in rural populations: the namil study in south korea. *American journal of ophthalmology*, 154(1):99–106, 2012.
- [235] B. Sweetman and A. Linniger. Cerebrospinal fluid flow dynamics in the central nervous system. *Annals of biomedical engineering*, 39(1):484–496, 2011.
- [236] K. M. Tangen, Y. Hsu, D. C. Zhu, and A. A. Linninger. Cns wide simulation of flow resistance and drug transport due to spinal microanatomy. *Journal of biomechanics*, 48(10):2144–2154, 2015.
- [237] Y.-C. Tham, S.-H. Lim, P. Gupta, T. Aung, T. Y. Wong, and C.-Y. Cheng. Inter-relationship between ocular perfusion pressure, blood pressure, intraocular pressure profiles and primary open-angle glaucoma: the singapore epidemiology of eye diseases study. *British Journal of Ophthalmology*, 102(10):1402–1406, 2018.
- [238] F. Topouzis, A. L. Coleman, A. Harris, C. Jonescu-Cuypers, F. Yu, L. Mavroudis, E. Anastasopoulos, T. Pappas, A. Koskosas, and M. R. Wilson. Association of blood pressure status with the optic disk structure in non-glaucoma subjects: the thessaloniki eye study. *American journal of ophthalmology*, 142(1):60–67, 2006.
- [239] M. W. Tuck and R. P. Crick. The age distribution of primary open angle glaucoma. *Ophthalmic epidemiology*, 5(4):173–183, 1998.
- [240] K. Tuma, J. Stein, V. PRUŠA, and E. Friedmann. Motion of the vitreous humour in a deforming eye–fluid–structure interaction between a nonlinear elastic solid and a nonlinear viscoelastic fluid. *arXiv preprint arXiv:1802.04002*, 2018.

- [241] M. Ursino and C. A. Lodi. A simple mathematical model of the interaction between intracranial pressure and cerebral hemodynamics. *Journal of Applied Physiology*, 82(4):1256–1269, 1997.
- [242] T. S. Vajaranant, S. Nayak, J. T. Wilensky, and C. E. Joslin. Gender and glaucoma: what we know and what we need to know. *Current opinion in ophthalmology*, 21(2):91, 2010.
- [243] J. Vardakis, D. Chou, B. Tully, C. Hung, T. Lee, P. Tsui, and Y. Ventikos. Investigating cerebral oedema using poroelasticity. *Medical engineering and physics*, 2016.
- [244] A. C. Verticchio Vercellin, A. Harris, T. A. Ciulla, G. Chiaravalli, R. Sacco, B. A. Siesky, I. Januleviciene, and G. Guidoboni. Quantifying retinal and choroidal contributions to macular oxygenation: a theoretical approach. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [245] B. Wang, H. Tran, M. A. Smith, T. Kostanyan, S. E. Schmitt, R. A. Bilonick, N.-J. Jan, L. Kagemann, E. C. Tyler-Kabara, H. Ishikawa, et al. In-vivo effects of intraocular and intracranial pressures on the lamina cribrosa microstructure. *PloS one*, 12(11):e0188302, 2017.
- [246] D. Wang, W. Huang, Y. Li, Y. Zheng, P. J. Foster, N. Congdon, and M. He. Intraocular pressure, central corneal thickness, and glaucoma in chinese adults: the liwan eye study. *American journal of ophthalmology*, 152(3):454–462, 2011.
- [247] N. Wang. Intraocular and intracranial pressure gradient in glaucoma.
- [248] R. N. Weinreb and A. Harris. *Ocular blood flow in glaucoma*, volume 6. Kugler Publications, 2009.
- [249] S. Whitaker. *The method of volume averaging*, volume 13. Springer Science & Business Media, 2013.
- [250] R. M. Wiard, H. J. Kim, C. A. Figueroa, G. T. Kovacs, C. A. Taylor, and L. Giovangrandi. Estimation of central aortic forces in the ballistocardiogram under rest and exercise conditions. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 2831–2834. IEEE, 2009.
- [251] T. H. Williamson and A. Harris. Color doppler ultrasound imaging of the eye and orbit. *Survey of ophthalmology*, 40(4):255–267, 1996.
- [252] G. Wollensak and E. Spoerl. Biomechanical characteristics of retina. *Retina*, 24(6):967–970, 2004.
- [253] S.-Y. Woo, A. Kobayashi, W. Schlegel, and C. Lawrence. Nonlinear material properties of intact cornea and sclera. *Experimental eye research*, 14(1):29–39, 1972.
- [254] J. D. Wright, J. P. Hughes, Y. Ostchega, S. S. Yoon, T. Nwankwo, et al. Mean systolic and diastolic blood pressure in adults aged 18 and over in the united states, 2001–2008. *Natl Health Stat Report*, 35(1-22):24, 2011.
- [255] D. Yan, F. Coloma, A. Metheetrairut, G. Trope, J. Heathcote, and C. Ethier. Deformation of the lamina cribrosa by elevated intraocular pressure. *British Journal of Ophthalmology*, 78(8):643–648, 1994.

- [256] D. Yan, J. Flanagan, J. Flanagan, T. Farra, G. Trope, and C. Ethier. Study of regional deformation of the optic nerve head using scanning laser tomography. *Current eye research*, 17(9):903–916, 1998.
- [257] S. J. Young, W. M. Gillon, R. V. Rifredi, and W. T. Krein. Method and apparatus for monitoring vital signs remotely, Mar. 27 2008. US Patent App. 11/849,051.
- [258] M. Žáček and E. Krause. Numerical simulation of the blood flow in the human cardiovascular system. *Journal of Biomechanics*, 29(1):13–20, 1996.
- [259] P. Zang, J. Wang, T. Hormel, L. Liu, D. Huang, and Y. Jia. Automated segmentation of peripapillary retinal boundaries in oct combining convolutional neural network and graph search. In *Annual Meeting of Association for Research in Vision and Ophthalmology (ARVO)*, 2019.
- [260] S. Zhao, X. Xu, and M. Collins. The numerical analysis of fluid-solid interactions for blood flow in arterial structures part 2: development of coupled fluid-solid algorithms. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 212(4):241–252, 1998.
- [261] E. Zimlichman, M. Szyper-Kravitz, Z. Shinar, T. Klap, S. Levkovich, A. Unterman, R. Rozenblum, J. M. Rothschild, H. Amital, and Y. Shoenfeld. Early recognition of acutely deteriorating patients in non-intensive care units: Assessment of an innovative monitoring technology. *Journal of Hospital Medicine*, 7(8):628–633, 2012.

Presentations and publications during the period of doctoral studies

Oral presentations

- **The Ocular Mathematical Virtual Simulator: towards uncertainty quantification.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
6th International Conference on Computational and Mathematical Biomedical Engineering. Sendai, Japan. 11/06/2019.
- **An operator splitting method for the time discretization of a multi-scale model in ophthalmology.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
9e Biennale Française des Mathématiques Appliquées et Industrielles. Guidel Plages (Morbihan), France. 14/05/2019.
- **A web-based interface for ocular hemodynamics and biomechanics analysis via the Ocular Mathematical Virtual Simulator.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS, A.C. VERTICCHIO VERCELLIN, B.A. SIESKY, A. HARRIS.
Annual Meeting of the Association for Research in Vision and Ophthalmology. Vancouver, Canada. 01/05/2019.
- **Mathematical modeling and simulation of ocular blood flow and their interactions.**
L. SALA.
Workshop Modeling the eye as a window on the body. American Institute of Mathematics, San José (CA), USA. 17/10/2018.
- **Ocular Mathematical Virtual Simulator: A Hemodynamical and Biomechanical Study towards Clinical Applications.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
13th World Congress on Computational Mechanics: Multidisciplinary Alliance in Biosciences: Modeling, Computing, Technology and Clinical Applications. New York (NY), USA. 24/07/2018.
- **Unconditionally stable operator splitting method for a multiscale application in ophthalmology.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
44e Congrès National d'Analyse Numérique. Centre Azureva, Cap d'Agde, France. 30/05/2018.
- **Analysis of IOP and CSF alterations on ocular biomechanics and lamina cribrosa hemodynamics.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS, B.A. SIESKY, A. HARRIS.
Annual Meeting of the Association for Research in Vision and Ophthalmology. Honolulu (HI), USA. 01/05/2018.

- **Towards a full model for ocular biomechanics, fluid dynamics and hemodynamics.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
ESB-ITA VII Annual Meeting. Roma, Italy. 28/09/2017.
- **Multiphysics modeling of the Eye using Feel++.**
L. SALA
Workshop 5th Feel++ User Days. IRMA, Strasbourg, France. 14/09/2017.
- **HDG Method and Toolbox in Feel++.**
R. HILD, L. SALA.
Workshop 5th Feel++ User Days. IRMA, Strasbourg, France. 13/09/2017.
- **Demo tissue perfusion in the eye.**
L. SALA.
Workshop Mathematical Modelling, Simulation and Optimization for Societal Challenges with Scientific Computing (MSO4SC). Budapest, Hungary. 23/05/2017.
- **Hi-POD reduction techniques for parametrized fluid dynamics problems.**
L. SALA.
Séminaire Equations aux dérivées partielles". IRMA, Strasbourg, France. 11/10/2016.

Poster presentations

- **Mathematical modeling and simulation of ocular blood flows and their interactions.**
L. SALA.
Poster Day: University of Strasbourg - Doctoral School in Mathematics, Engineering and Computer Science. Pôle API, Illkirch-Graffenstaden, France. 02/10/2017.
- **Patient-specific virtual simulator of tissue perfusion in the lamina cribrosa.**
L. SALA, C. PRUD'HOMME, D. PRADA, F. SALERNI, C. TROPHIME, V. CHABANNES, M. SZOPOS, R. REPETTO, S. BERTOLUZZA, R. SACCO, A. HARRIS, G. GUIDOBONI.
Annual Meeting of the Association for Research in Vision and Ophthalmology. Baltimore(MD), USA. 07/05/2017.
- **Hi-POD reduction techniques for parametrized fluid dynamics problems.**
L. SALA, D. BAROLI, C.M. COVA, S. PEROTTO, A. VENEZIANI.
Numerical methods for PDEs: recent developments in numerical methods for model reduction. IHP, Paris, France. 08/11/2016.

Patents

- **Cardiovascular function and ballistocardiogram.** US provisional patent application No.62/735,716.
Filed on 09/24/2018.
Inventor: G. Guidoboni.
Authors: G. Guidoboni, L. Sala.
Assignee: The Curators of the University of Missouri.

Peer-reviewed articles in International Journals

- **Energy-based operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows: the Darcy case.**
L. SALA, G. GUIDOBONI, M. SZOPOS, C. PRUD'HOMME.
In preparation.
- **The Ocular Mathematical Virtual Simulator: preliminary sensitivity analysis.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
In preparation.

- **The Ocular Mathematical Virtual Simulator: validation strategy.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS, A.C. VERTICCHIO VERCELLIN, B.A. SIESKY, A. HARRIS.
In preparation.
- **HDG methods with integral boundary conditions: theory and applications.**
S. BERTOLUZZA, G. GUIDOBONI, R. HILD, D. PRADA, C. PRUD'HOMME, R. SACCO, L. SALA, AND M. SZOPOS.
In preparation.
- **Mathematical modeling of the cerebrospinal fluid flow and its interactions.**
L. SALA, F. SALERNI, M. SZOPOS.
Chapter in the book Mathematical Modeling of Ocular Fluid Dynamics: From Theory to Clinical Applications.
Editors: G. Guidoboni, A. Harris, R. Sacco.
Springer-Birkhauser (New York). *Book series:* Modeling and Simulation in Science, Engineering, and Technology *Expected release date:* July 1, 2019.
- **A Theoretical Study of Aqueous Humor Secretion Based on a Continuum Model Coupling Electrochemical and Fluid-Dynamical Transmembrane Mechanisms.**
L. SALA, A.G. MAURI, R. SACCO, D. MESSENIO, G. GUIDOBONI, A. HARRIS.
Communications in Applied Mathematics and Computational Science. 2019.
- **Cardiovascular function and ballistocardiogram: a relationship interpreted via mathematical modeling.**
G. GUIDOBONI, L. SALA, M. ENAYATI, R. SACCO, M. SZOPOS, J.M. KELLER, M. POPESCU, L. DESPINS, V. HUXLEY, M. SKUBIC.
IEEE Transactions on Biomedical Engineering. 2019
- **Towards a full model for ocular biomechanics, fluid dynamics, and hemodynamics.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
Journal for Modeling in Ophthalmology, 2(2), pp.7-13. 2018.
- **Ocular mathematical virtual simulator: A hemodynamical and biomechanical study towards clinical applications.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
Journal of Coupled Systems and Multiscale Dynamics, 6(3), pp.241-247. 2018
- **Multiscale nature of ocular physiology.**
L. SALA, R. SACCO, G. GUIDOBONI.
Journal for Modeling in Ophthalmology, 2(1), pp.12-18. 2018.

Peer-reviewed proceedings in National and International Conferences

- **The Ocular Mathematical Virtual Simulator: towards uncertainty quantification.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS.
Proceedings of the 6th International Conference on Computational and Mathematical Biomedical Engineering, 2(1), pp. 429-432. June, 2019.
- **A web-based interface for ocular hemodynamics and biomechanics analysis via the Ocular Mathematical Virtual Simulator.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS, A.C. VERTICCHIO VERCELLIN, B.A. SIESKY, A. HARRIS.
Accepted.
- **Analysis of IOP and CSF alterations on ocular biomechanics and lamina cribrosa hemodynamics.**
L. SALA, C. PRUD'HOMME, G. GUIDOBONI, M. SZOPOS, B.A. SIESKY, A. HARRIS.
Investigative Ophthalmology & Visual Science, 59(9), pp.4475-4475. 2018.

- **A theoretical study of the role of conformational properties of transepithelial ion pumps on aqueous humor production.**
R. SACCO, L. SALA, A.G. MAURI, D. MESSENIO, G. GUIDOBONI, B.A. SIESKY, A. HARRIS
Investigative Ophthalmology & Visual Science, 59 (9), 1656-1656. 2018
- **A theoretical study of the role of conformational properties of transepithelial ion pumps on aqueous humor production.**
R. SACCO, A.G. MAURI, L. SALA, S. CASSANI, B.A. SIESKY, G. GUIDOBONI, A. HARRIS.
Investigative Ophthalmology & Visual Science, 59(9), pp.1656-1656. 2018
- **Unconditionally stable operator splitting method for a multiscale application in ophthalmology.**
G. GUIDOBONI, C. PRUD'HOMME, L. SALA, M. SZOPOS.
Proceedings of the 44e Congrès National d'Analyse Numérique. May, 2018.
- **Patient-specific virtual simulator of tissue perfusion in the lamina cribrosa.**
L. SALA, C. PRUD'HOMME, D. PRADA, F. SALERNI, C. TROPHIME, V. CHABANNES, M. SZOPOS, R. REPETTO, S. BERTOLUZZA, R. SACCO, A. HARRIS.
Proceedings of the Annual Meeting of Association for Research in Vision and Ophthalmology. May, 2017.

Optic neuropathies such as glaucoma are often late-onset, progressive and incurable diseases. Despite the recent progress in clinical research, there are still numerous open questions regarding the etiology of these disorders and their pathophysiology. Furthermore, data on ocular posterior tissues are difficult to estimate noninvasively and their clinical interpretation remains challenging due to the interaction among multiple factors that are not easily isolated. The recent use of mathematical models applied to biomedical problems has helped unveiling complex mechanisms of the human physiology.

In this very compelling context, our contribution is devoted to designing a mathematical and computational model coupling tissue perfusion and biomechanics within the human eye. In this thesis we have developed a patient-specific Ocular Mathematical Virtual Simulator (OMVS), which is able to disentangle multiscale and multiphysics factors in a accessible environment by employing advanced and innovative mathematical models and numerical methods. Moreover, the proposed framework may serve as a complementary method for data analysis and visualization for clinical and experimental research, and a training application for educational purposes.

In the first part of the thesis, we describe the anatomy of the eye and the pathophysiology of glaucoma. Next, we define the modeling choices and the mathematical architecture of the OMVS (Part II). In part III we present the complex ocular geometry and mesh along with the new numerical methods we have developed, namely the Hybridizable Discontinuous Galerkin method with Integral Boundary Conditions, and the operator splitting technique for solving coupled PDE-ODE systems. The fourth part of the thesis gathers all the C++ libraries that have been implemented to create and solve the OMVS. Part V illustrates the OMVS simulation results, specifically the verification and the validation strategy, and some clinically meaningful virtual experiments. Then, we propose a preliminary uncertainty quantification study, in particular an analysis on the propagation of uncertainties and a sensitivity analysis using Sobol indices (Part VI). Finally, in the last part of the thesis, we draw the conclusions and characterize different projects that can be integrated in the OMVS in the future.



cemosis
Centre de modélisation et de simulation de Strasbourg

INSTITUT DE RECHERCHE MATHÉMATIQUE AVANCÉE
UMR 7501
Université de Strasbourg et CNRS
7 Rue René Descartes
67 084 STRASBOURG CEDEX



Université
de Strasbourg

Tél. 03 68 85 01 29
Fax 03 68 85 03 28
irma.math.unistra.fr
irma@math.unistra.fr






IRMA
Institut de Recherche
Mathématique Avancée

IRMA 2019/008
<http://tel.archives-ouvertes.fr/???>

ISSN 0755-3390