

# Mathematical and Computational Modeling of Biodegradation Behavior of Personalized Printed Implants

**Mojtaba Barzegari Shankil**

Supervisor:  
Prof. dr. ir. L. Geris

Dissertation presented in partial  
fulfillment of the requirements for the  
degree of Doctor of Engineering  
Science (PhD): Mechanical  
Engineering

November 2022



# **Mathematical and Computational Modeling of Biodegradation Behavior of Personalized Printed Implants**

**Mojtaba BARZEGARI SHANKIL**

Examination committee:

Prof. dr. ir. The Chairman, chair  
Prof. dr. ir. L. Geris, supervisor  
Prof. dr. ir. H. Van Oosterwyck  
Prof. dr. ir. N. Moelans  
Prof. dr. G. Závodszyk  
(University of Amsterdam)

Dissertation presented in partial  
fulfillment of the requirements for  
the degree of Doctor of Engineering  
Science (PhD): Mechanical Engineering

November 2022

© 2022 KU Leuven – Faculty of Engineering Science  
Uitgegeven in eigen beheer, Mojtaba Barzegari Shankil, Celestijnenlaan 300 box 2419, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

# Preface

Here we will have acknowledgment, in which I talk about people supported me during my PhD.

سلام این یک تست است و من آن را برای شما نوشته ام. امیدوارم این نوشته ازاردنه نباشد. این متن برای قدردانیست و ممکن است کمی طولانی شود.

This is the continuation of the text.



# **Abstract**

...



# **Beknopte samenvatting**

...



# List of Abbreviations

**MD** molecular dynamics. 6



# List of Symbols

$\Theta$  A nice symbol



# Contents

<b>Abstract</b>	iii
<b>Beknopte samenvatting</b>	v
<b>List of Abbreviations</b>	vii
<b>List of Symbols</b>	ix
<b>Contents</b>	xi
<b>List of Figures</b>	xv
<b>List of Tables</b>	xvii
<b>1 Introduction</b>	1
1.1 Chemistry of biodegradation of magnesium . . . . .	1
1.1.1 Degradation rate evaluation techniques . . . . .	5
<b>2 Aim and objective</b>	7
<b>3 Developing the core computational model</b>	9
3.1 Introduction . . . . .	9
3.1.1 Magnesium biodegradation . . . . .	9
3.1.2 Computational modeling of Mg degradation . . . . .	10
3.2 Background theory . . . . .	12
3.2.1 Biodegradation as a reaction-diffusion system . . . . .	12
3.2.2 Moving boundary - Stefan problems . . . . .	13
3.2.3 Level set method . . . . .	14
3.3 Materials and methods . . . . .	14

3.3.1	Underlying chemistry . . . . .	14
3.3.2	Mathematical modeling . . . . .	15
3.3.3	Interface movement formulation . . . . .	17
3.3.4	Boundary conditions . . . . .	18
3.3.5	Implementation . . . . .	19
3.3.6	Experimental setup . . . . .	20
3.3.7	Parameter estimation . . . . .	21
3.3.8	Simulation setup . . . . .	23
3.3.9	Case study . . . . .	25
3.4	Results . . . . .	25
3.4.1	Optimization results . . . . .	25
3.4.2	Degradation prediction . . . . .	26
3.4.3	Example application . . . . .	28
3.5	Discussion . . . . .	29
3.6	Conclusions . . . . .	33
<b>4</b>	<b>Fluid flow and convection</b>	<b>35</b>
<b>5</b>	<b>Advanced model with kinetics</b>	<b>37</b>
<b>6</b>	<b>Tissue growth</b>	<b>39</b>
<b>7</b>	<b>Model parallelization</b>	<b>41</b>
7.1	Introduction . . . . .	41
7.2	Background theory and model description . . . . .	44
7.2.1	Chemistry of degradation . . . . .	45
7.2.2	Reaction-diffusion equation . . . . .	45
7.2.3	Level set method . . . . .	47
7.3	Methodology of model implementation . . . . .	49
7.3.1	Finite element discretization (bottleneck of the algorithm)	50
7.3.2	Implementation and parallelization . . . . .	54
7.3.3	Level set issues . . . . .	57
7.3.4	Simulation setup . . . . .	59
7.3.5	Performance analysis . . . . .	60
7.3.6	Compute environment . . . . .	62
7.4	Results . . . . .	62
7.4.1	Numerical simulation results . . . . .	62
7.4.2	Weak and strong scaling results . . . . .	63
7.5	Discussion . . . . .	65
7.6	Conclusion . . . . .	68

7.A Comparing the performance of different combinations of KSP types and preconditioners . . . . .	69
<b>8 BioDeg software</b>	<b>71</b>
8.1 Summary . . . . .	71
8.2 Statement of need . . . . .	72
8.3 Overview of BioDeg . . . . .	73
8.3.1 Basic concepts . . . . .	73
8.3.2 Implementation . . . . .	74
8.3.3 High-performance computing . . . . .	76
8.3.4 Installation . . . . .	76
8.3.5 Documentation . . . . .	77
8.3.6 Contribution . . . . .	77
8.3.7 Published works . . . . .	78
8.4 Acknowledgements . . . . .	78
<b>9 Bayesian parameter estimation of the computational models</b>	<b>79</b>
9.1 Summary . . . . .	79
9.2 Statement of Need . . . . .	81
9.3 Learning objectives . . . . .	81
9.4 Prerequisites . . . . .	82
9.5 Pedagogy and instructional design . . . . .	82
9.6 Getting started . . . . .	83
9.7 Acknowledgements . . . . .	83
9.A A glimpse of the Jupyter notebook . . . . .	84
9.A.1 Introduction . . . . .	84
9.A.2 Bayesian optimization . . . . .	85
9.A.3 Sample problem . . . . .	87
<b>10 Applications: Mandible plate</b>	<b>89</b>
<b>11 Applications: Acetabular cup</b>	<b>91</b>
<b>12 Applications: Mechanical integrity of infilled structures</b>	<b>93</b>
<b>13 Conclusion</b>	<b>95</b>
<b>Bibliography</b>	<b>97</b>
<b>This is curriculum</b>	<b>109</b>



# List of Figures

1.1	Commonly used electrolytes for testing biodegradable metals . . . . .	2
1.2	Mg biodegradation behavior in commonly used test solutions . . . . .	6
3.1	Simplified chemistry of biodegradation of Mg . . . . .	16
3.2	Schematic overview of exposed boundary conditions and constraints for biodegradation simulations . . . . .	19
3.3	Simulation setup of the biodegradation tests . . . . .	24
3.4	Simulation results of the cuboid sample in the biodegradation test	27
3.5	Change of concentration of various chemical components over a line . . . . .	28
3.6	Simulation results of the sample screw in the biodegradation test	29
7.1	Schematic presentation of different components of the studied moving boundary problem . . . . .	44
7.2	The implicit function definition in the studied problem . . . . .	48
7.3	Overlapping domain decomposition . . . . .	56
7.4	Comparison of the sparsity patterns of the global matrix A . . . . .	56
7.5	Simulation setup for evaluating parallel performance . . . . .	60
7.6	Models used for performing the weak-scaling test . . . . .	61
7.7	Numerical simulation result of the reaction-diffusion system . . . . .	63
7.8	Weak-scaling test result . . . . .	64
7.9	Speed-up and parallel efficiency of the weak-scaling experiment	64
7.10	Strong-scaling test result . . . . .	65
7.11	Speed-up and parallel efficiency of the strong-scaling test . . . . .	65
7.12	Performance test result for various preconditioners and solvers .	70
8.1	Simulation result of degradation of a screw performed using BioDeg . . . . .	75

9.1 A typical output of the parameter estimation process . . . . .	88
--	----

# List of Tables

1.1	Summary of various common methods to asses the degradation rate of Mg . . . . .	6
3.1	Chemical composition of NaCl and SBF electrolytes . . . . .	22
3.2	Effective parameters to be considered in the Bayesian optimization for parameter calibration . . . . .	26
3.3	Values used in the biodegradation model simulation performance	26
7.1	Strong-scaling test result, presented by the execution time of each PDE in simulations with a different number of employed MPI cores. . . . .	64
7.2	Performance test result for various combinations of preconditioners and solvers, presented by the execution time of each PDE.	70



# **Chapter 1**

## **Introduction**

The application of biodegradable metallic biomaterials [1, 2, 3], including magnesium [4, 5, 6], zinc [7, 8], and iron [9], has become more prominent for over a decade in various biomedical engineering and tissue engineering disciplines. Among the mentioned materials, magnesium (Mg) is the most studied metals [10], the reason of which is its suitable mechanical and chemical properties for biomedical applications. Although poor corrosion resistance of Mg is a limiting factor for its application as a light structural material, like in transportation industry, it becomes an interesting characteristic when it comes to biodegradable materials field for cardiovascular and orthopedic applications [11, 12, 13]. The first clinical usage of Mg was reported in 1878, but a renewed interest on it has grown significantly in the last 15-20 years [10]. From the clinical and biomedical perspective, two major concerns about using Mg in clinics are the release of hydrogen gas and surface alkalization due to Mg dissolution [14]. These issues are commonly addressed by alloying, biocompatible coating, and surface modification [10].

### **1.1 Chemistry of biodegradation of magnesium**

The biodegradation behavior of Mg is investigated in corrosion tests, in which the selection of the corrosive media plays an important role since it affects the underlying chemical reactions [15]. By considering the main application

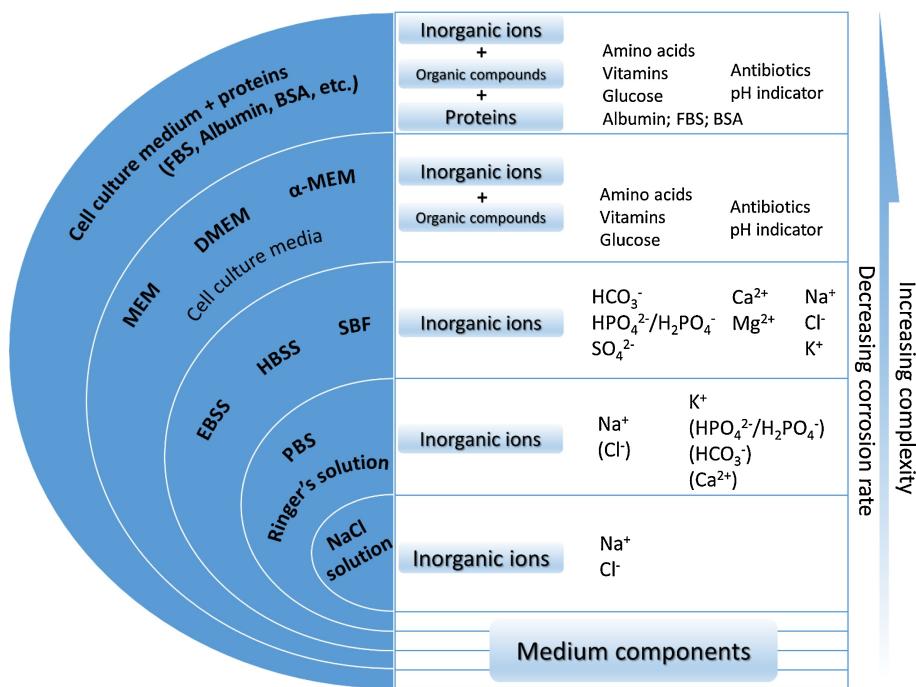


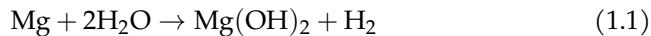
Figure 1.1: A schematic representation of commonly used electrolytes for testing biodegradable metals, sorted by their complexity from the chemical perspective from bottom to top [15].

of the biomaterial, which can be tissue engineering scaffolds, vascular stents, or orthopedic fixation implants, the corrosive media can be selected to be a representative of the service environment. The most basic form of the medium is a saline (NaCl) solution, in which the degradation rate is the highest possible [15]. More complex solutions can be used to mimic the behavior of body environment by taking into account more body fluid components, the most popular of which are the Ringer's solution, PBS (phosphate buffered saline), SBFs (simulated body fluids), HBSS (Hank's balanced salt solution), and Earle's balanced salt solution (EBSS) [15]. Adding more organic components to the solution will make it ready to simulate cell culture conditions. The common media for this purpose are MEM (Minimum Essential medium) and DMEM (Dulbecco's modified Eagle's medium) [15]. Fig. 1.1 summarizes various commonly used electrolytes for testing biodegradable metals [15].

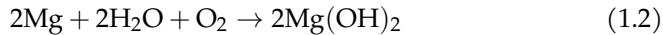
A wide range of various studies have already investigated the effect of different components in the aforementioned corrosive media on the degradation behavior of Mg materials [16, 17, 18, 19, 20]. In addition to the presented chemical components, synthetic pH buffers, such as Tris as HEPES), has been shown to contribute to the biodegradation rate of Mg [16]. The performed investigations on the effect of different inorganic components, including carbonate, phosphate, sulfate, and calcium, show the effective contribution of these components on the rate of degradation, although the corrosion protection resulted from the mutual effect of carbonate, phosphate and calcium has been more emphasized [16, 19].

The most common solution for performing corrosion test on Mg is saline (NaCl) solution, in which the material undergoes an aggressive corrosion due to higher electrochemical activities [21, 22]. In a typical aqueous solution, the major corrosion reactions occurring can be written as [23, 24]:

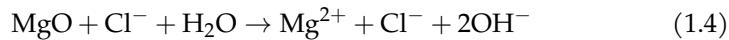
Main, hydrogen evolution reaction (HER):



Secondary, oxygen reduction reaction (ORR):



In this situation, the corrosion products forming on the corroded surface of Mg consists mainly of  $\text{Mg(OH)}_2$  and  $\text{MgO}$ , and the pH in regions close to this surface remains alkaline. In presence of chloride ions in the saline medium, the formed corrosion product may be broken or bypassed, leading to increased degradation rate:



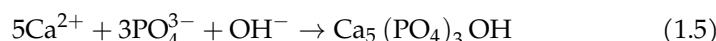
The main advantage of using a saline solution for corrosion tests in comparison to more complex media is that absence of inorganic ions like carbonate, phosphate, sulfate, and calcium allows investigating the corrosion behavior without concerning possible contamination by microbial life. On the other hand, the main weakness of saline solution is that it cannot represent the

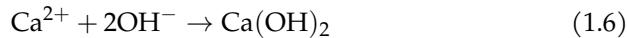
complexity of real body fluid, and as a result, a more complex medium is required to investigate such conditions. To address this issue, more complex saline solutions, such as PBS are widely used for assessing the applicability of Mg alloys in more complex conditions from the chemical perspective [25, 26]. Despite the mentioned limitations, corrosion test in saline solution is beneficial for understanding intrinsic degradation properties of Mg.

The term "simulated body fluid" is generally used to refer to solutions containing inorganic ions of human serum and interstitial fluid in corrosion tests [15]. The commonly used media in this regard are SBF, HBSS, and EBSS, which include the same inorganic components with a slight difference in their concentrations. A typical composition of these media is chloride, carbonate, phosphates, sulfate, and calcium. The individual effect of these components on the rate of degradation of Mg has been extensively studied, in which it has been observed that carbonate and phosphate slow down the rate while the effect of sulfate is negligible [18, 20]. The concentration of  $\text{HCO}_3^-$  affects the pH buffering capacity and the degradation rate of Mg simultaneously [27].

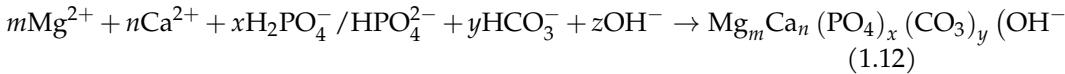
The effect of calcium ion is more complex because it has been found that  $\text{Ca}^{2+}$  doesn't contribute to the Mg corrosion directly [6], but a mixed effect of  $\text{Ca}^{2+}$ ,  $\text{Mg}^{2+}$ ,  $\text{HCO}_3^-$ , and  $\text{H}_2\text{PO}_4^-/\text{HPO}_4^{2-}$  forms a co-precipitation layer on the corroded surface of Mg, slowing down the corrosion rate of commercially pure Mg as well as some of the Mg alloys [16, 19]. It has been also reported that although various Mg alloys show different intrinsic degradation behavior in NaCl solution, they possibly behave similarly in simulated body fluids [28, 20]. Since the humoral regulations inside the human body control the changes in pH of body fluids, it's common to use pH buffers to mimic a similar condition, but it should be taken into account that buffering solution may affect the degradation rate of Mg [29, 30]. pH buffers may also delay the formation of the precipitate layer [19]. An alternative solution to address this issue is to use natural pH buffers such as  $\text{HCO}_3^-/\text{CO}_2$ , a technique that is commonly used for immersion tests under cell culture conditions. In this situation, an equilibrium between  $\text{H}_2\text{CO}_3(\text{CO}_2)$ ,  $\text{HCO}_3^-$ , and  $\text{CO}_3^{2-}$  keeps the pH constant. As a result, using simulated body fluids for corrosion tests without additional synthetic pH buffer is still acceptable [19, 20].

The major reactions occurring in simulated body fluids can be written as:





Then, the aforementioned protection layer is formed on the corroded surface as a hydroxyapatite-like precipitation according the following reaction [24, 31, 32, 33]:



In fact, the similarity in corrosion behavior of various Mg alloys in SBF solutions is because of the similar composition of this quasi-protective layer, a mechanism that doesn't occur in NaCl solution, leading to more apparent difference of degradation rate between Mg alloys. The composition of the formed hydroxyapatite-like precipitation layer is close to the ones found *in vivo* [15]. Additionally, local pH measurements in HBSS and SBF shows that in contrast to saline solutions, the local pH value is not alkaline [19, 34]. This has been reported for hydrodynamics situation, under which the medium composition is kept constant by replacing the consumed ions by means of fluid flow. Fig. 1.2 briefly summarizes the various reactions and formed precipitation compositions of the mentioned media for testing the degradation behavior of Mg [15].

### 1.1.1 Degradation rate evaluation techniques

Generally speaking, the method used for evaluating the degradation rate can affect the reported behavior. For example, it has been shown that in HBSS, the measured corrosion rate of Mg is lower (slower) when evaluated using hydrogen evolution in comparison to the rate found by direct weight loss measurements [35, 36], which can be due to the secondary dissolution of evolved hydrogen. Moreover, the consumption of oxygen due to secondary ORR can affect the volume of evolved gas, which is more significant for

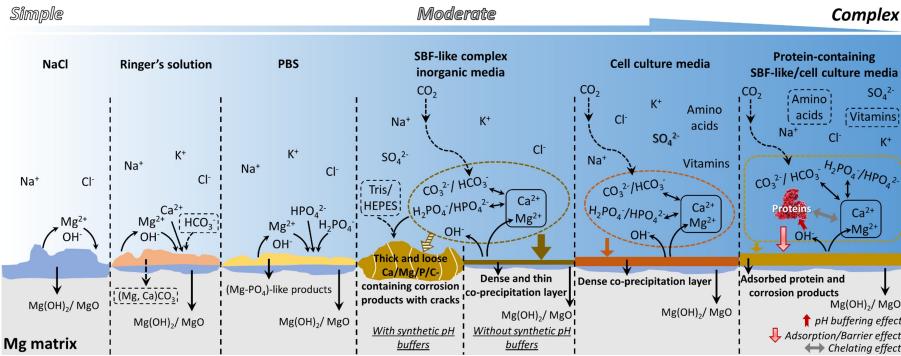


Figure 1.2: A schematic representation of Mg biodegradation behavior in commonly used solutions for corrosion tests of biodegradable metals [15].

media with slower degradation rate such as HBSS and MEM [37]. Table 1.1 summarizes the advantages and shortcomings of widely used techniques for measuring degradation rate [15].

Table 1.1: Summary of various common methods to asses the degradation rate of Mg

Test method	Advantages	Shortcomings
Weight loss	High reliability Direct measurement Easily controlled test environment	Non-continuous. Does not reveal varying corrosion rate throughout the immersion Low sensitivity at the initial stages
Hydrogen evolution	Continuous Can be automated Can be performed in closed eu-dimeters	Performed in open environment in most cases Might show underestimated values of corrosion rate due to secondary ORR and solubility of H <sub>2</sub> in aqueous media
Potentiodynamic polarization	Fast measurement	Non-continuous Open environment measurement in most cases Very often low correlation with long-term weight loss measurements
Electrochemical impedance spectroscopy	Continuous In situ investigation of protective properties of forming corrosion products	Performed in open environment in most cases

Introducing some symbol:  $\Theta$ .

Introducing an acronym: MD.

## **Chapter 2**

### **Aim and objective**



# Chapter 3

## Developing the core computational model

This chapter is based on previously published content in *Corrosion Science*: M. Barzegari, D. Mei, S. V. Lamaka, and L. Geris, “Computational modeling of degradation process of biodegradable magnesium biomaterials,” *Corrosion Science*, vol. 190, p. 109674, Sep. 2021.

### 3.1 Introduction

#### 3.1.1 Magnesium biodegradation

Due to their bio-friendly properties, biodegradable metallic biomaterials, including magnesium (Mg), iron (Fe), and zinc (Zn), are regaining attention in recent years [1]. These biomaterials find important applications in the design and manufacturing of supportive implants such as temporary devices in orthopedics and the cardiovascular field [38, 4]. In orthopedics, the biodegradable metallic biomaterials are used as fixation devices, providing adequate support in the early stages while being absorbed gradually during the bone healing process [39]. Implants fabricated using Mg and its alloys are being used for such a purpose [40] due to the similarity of the stiffness between

natural bone and Mg, which helps to reduce the stress shielding induced by the implanted device. Additionally, Mg is reported to have a non-toxic contribution to the human body's metabolism and the bone healing process, which makes the release and absorption of metallic ions safe and biocompatible [41].

Accumulation of mechanistic understanding of Mg degradation achieved by experimental approaches over the years gradually provided a mechanistic understanding of the biodegradation process. Combining these insights with *in silico* modeling approaches enables researchers to study the biodegradation properties and behavior of the implant in a virtual environment prior to conducting any *in vitro* or *in vivo* tests. When fully validated, computational modeling can (in part) replace certain stages of costly and time-consuming experiments verifying the expected degradation behavior of the designed implants. Additionally, the developed models can be efficiently combined with existing computational models to examine other related phenomena such as tissue growth or mechanical integrity.

### 3.1.2 Computational modeling of Mg degradation

Previous contributions to the computational modeling of the degradation process include a wide range of different approaches, from the basic phenomenological implementations to comprehensive mechanistic models that take into account various aspects of the degradation and resorption process.

Continuous damage (CD) modeling has always been a common approach for corrosion simulation, but from a physicochemical point of view, it focuses on the mechanical integrity of the degradation and neglects the diffusion process. As a result, its application in the degradation modeling of biomaterials, which includes various fundamental phenomena such as mass transfer through diffusion and reaction, is relatively limited. Despite this issue, a CD model proposed by Gastaldi et al. showed a good performance for simulation of bioresorbable Mg-based medical devices [42], in which geometrical discontinuities were interpreted as the reduction of material.

Alternatively, mathematical modeling using transport phenomena equations has shown great flexibility in capturing different mechanisms involved in the biodegradation process. As an example, in Ahmed et al., a set of mathematical equations in cylindrical and spherical coordinates was derived to model the chemical reactions of Mg degradation [43]. Despite the simplicity of

their approach from the computational perspective, their model was able to demonstrate the contribution of various chemical components to the *in vitro* degradation of Mg. Similarly, Grogan et al. developed a mathematical model based on the Stefan problem formulation in 1D space to correlate the mass flux of metallic ions into the solution to the velocity of shrinkage of the material during degradation [44]. This was done by considering the mass diffusion and change of the concentration of Mg<sup>2+</sup> ions, and then, employing an arbitrary Eulerian-Lagrangian (ALE) approach to extend the model to 3D on an adaptive mesh. A similar approach was taken by Shen et al. to develop a theoretical model of the degradation behavior of Mg-based orthopedic implants showing great consistency with *in vitro* test results [45].

An ultimate application of the computational modeling of the biodegradation process of biomaterials can be the prediction of how biodegradation affects the shape of the bulk material, medical device, or implant over time. One of the ways to achieve such a prediction is to capture the movement of the corrosion front mathematically using an appropriate method. The level set method (LSM) is a widely used example in this regard, which is an implicit mathematical way of representing the moving interfaces. This approach was used in Wilder et al. to study galvanic corrosion of metals [46]. They employed LSM on an adaptive mesh to track the moving corrosion interface, but their model lacked a thorough validation using experimental data. Gartzke et al. also worked on a simplified representation of the interface movement by developing a mechanochemical model of the biodegradation process, which helped them to study the effect of degradation on the mechanical properties [47]. They performed a basic qualitative validation on the predictions made by the model. Another similar study in this regard is the Sun et al. work [48], in which a detailed mathematical model was derived and validated to study the deposition of corrosion products on the surface of materials. This mathematical approach was also employed in the biomedical field by Bajger et al. to study the mass loss of Mg biomaterials during biodegradation [49]. They used LSM as well as a set of reaction-diffusion equations to track the change of geometry, which can be directly correlated to the loss of material. The derived equations were also able to capture the formation of the corrosion film that decreases the rate of degradation. Another comprehensive mathematical model was developed by Sanz-Herreraa et al. to investigate the role of multiple chemical components involved in the *in vitro* degradation of Mg implants [50]. One important drawback of this study was its 2D nature. Although the computational model was capable of studying the effect of multiple

components, due to the high number of derived equations, it would be difficult to extend and use the same model for real 3D implants. Additionally, a 2D model cannot capture the full phenomenon of corrosion, and as a result, the validation of the model will be more qualitative. It was shown in the study conducted by Gao et al. [51], where they compared the results of a multi-dimension model of the degradation of cardiovascular stents with those of a single-dimension model, that the number of considered dimensions had an important effect on the model predictions. In the end, it is worth mentioning that no dedicated experiments were performed in the aforementioned studies to validate the constructed mathematical and computational models.

The current study focuses on developing a physicochemical model of the biodegradation process of commercially-pure (CP) Mg biomaterials by continuing the work of Bajger et al. [49]. In this model, a set of partial differential equations (PDE) was derived according to the underlying chemistry of biodegradation, described as reaction-diffusion processes taking place at the interface of the biomaterial and its surrounding environment. The formation of a protective layer, effects of the ions in the solution, and the change in the pH due to the corrosion phenomenon were taken into account in the mathematical model. The corresponding computational model was implemented in a fully parallelized manner. Model calibration and validation were executed using data obtained from the immersion tests performed in saline (NaCl) and simulated body fluid (SBF) solutions.

## 3.2 Background theory

### 3.2.1 Biodegradation as a reaction-diffusion system

The biodegradation process can be considered as a reaction-diffusion system [52], in which the ions are released due to the chemical reactions on the surface, and the released ions diffuse through the surrounding solution and materials. These ions can interact with other ions and form new compounds [15]. As the reaction-diffusion systems have been studied in science and engineering for a couple of decades, the analogy with a reaction-diffusion system makes it convenient to construct a mathematical model of the biodegradation process based on the well-established transport phenomena equations [53]. From the mathematical perspective, a reaction-diffusion system is expressed by a set of

parabolic PDEs that describe the conservation of contributing chemical species in the studied system.

### 3.2.2 Moving boundary - Stefan problems

Moving boundary problems, also called Stefan problems, are the general class of mathematical problems in which the boundary of the domain should also be calculated in addition to the solution of the other equations [54]. Coupling the reaction-diffusion system of biodegradation with a moving boundary problem constructs a mathematical model in which the change of the domain geometry due to the material loss can be correlated to the underlying reaction and diffusion processes of corrosion. As the geometry can be determined accurately, this approach provides a way to measure the mass loss directly by computing the change in the volume of the material. In such a system, the moving boundary is the material-solution interface (corrosion front).

For a 1D corrosion diffusion system, the position of the diffusion interface can be determined by [54]:

$$s(t) = s_0 + 2\alpha \sqrt{t}, \quad (3.1)$$

in which the  $s(t)$  represents the position at any given time, and  $s_0$  is the initial interface position.  $\alpha$  coefficient can be calculated using:

$$\alpha = \frac{[\text{Mg}]_0 - [\text{Mg}]_{\text{sat}}}{[\text{Mg}]_{\text{sol}} - [\text{Mg}]_{\text{sat}}} \sqrt{\frac{D}{\pi}} \frac{\exp\left(\frac{-\alpha^2}{D}\right)}{\text{erfc}\left(\frac{-\alpha}{\sqrt{D}}\right)} \quad (3.2)$$

where  $[\text{Mg}]_{\text{sol}}$  is the concentration in the solid bulk (i.e. materials density) and  $[\text{Mg}]_{\text{sat}}$  is the concentration at which the material is released to the medium.  $[\text{Mg}]_0$  represents the initial concentration of the metallic ions in the medium, which is usually zero for most corrosion cases.

Eqs. 3.1 and 3.2 can be used to simply track the movement of the corrosion front, which is the employed method in studies like the Gorgan et al. work [44], but apparently, the real-world corrosion problems are 3D and much more complex than the described system.

As will be described later, Eq. 3.1 is used strictly for the first time step of the simulations in low diffusion regimes for calculating the initial velocity of the

interface. Generally speaking, a more sophisticated approach, such as the level set method, is required for tracking the interface of complex 3D geometries.

### 3.2.3 Level set method

In the current study, the corrosion front is tracked using an implicit function such that the zero iso-contour of the function represents the metal-solution interface. As a common practice, this implicit function is expressed as a signed distance function that defines the distance of each point of space (the domain of interest) to the interface. Such a definition implies that the zero iso-contour of the function belongs to the interface. The level set method provides an equation to declare such an implicit function,  $\phi = \phi(\mathbf{x}, t)$ ,  $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ , which can be obtained by solving [55]:

$$\frac{\partial \phi}{\partial t} + \vec{V^E} \cdot \nabla \phi + V^N |\nabla \phi| = b\kappa |\nabla \phi| \quad (3.3)$$

in which  $\vec{V^E}$  is the external velocity field, and  $V^N$  is the value of the normal interface velocity. The last term is related to the curvature-dependent interface movement and is omitted. As the effect of perfusion is neglected in the current study, the term containing the external velocity is also eliminated, resulting in the following simplified form of the level set equation:

$$\frac{\partial \phi}{\partial t} + V^N |\nabla \phi| = 0. \quad (3.4)$$

By having the normal velocity of the interface ( $V^N$ ) at each point and solving Eq. 3.4, the interface can be captured at the zero iso-contour of the  $\phi$  function.

## 3.3 Materials and methods

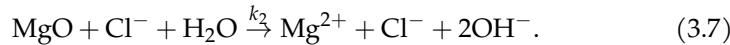
### 3.3.1 Underlying chemistry

The chemistry of biodegradation of Mg depends considerably on the surrounding solution and the presence of certain ions [15]. In NaCl solutions, the anodic and cathodic reactions as well as the formation and elimination of side

corrosion products can be considered as follows [1]:



Reaction 3.6 is not fully correct from the chemical point of view. In fact, Mg surface is always covered by MgO layer, and Mg(OH)<sub>2</sub> forms on top of that either at atmospheric conditions or during the immersion. The integrity of this MgO layer is undermined by Cl<sup>-</sup> ions, leading to an increase in degradation rate:



Although Cl<sup>-</sup> formally does not participate in reaction 3.7, it reflects the dependence of Mg corrosion rate on Cl<sup>-</sup> concentration. This effect on the rate of degradation has been widely expressed as the effect of Cl<sup>-</sup> on the Mg(OH)<sub>2</sub> in the literature [1, 4]. In the developed model, this effect is used interchangeably by omitting the MgO component, so the protective film formed on the corrosion interface is assumed to contain Mg(OH)<sub>2</sub> only. Moreover, it has been shown recently that oxygen reduction reaction also takes place during corrosion of Mg [37, 56, 32]. However, this is a secondary reaction (complementing water reduction) contributing to 1-20% of the total cathodic current depending on the conditions. Hence, it is not taken into consideration in this model. Additionally, the involved chemical reactions are more complicated in SBF solutions due to the presence of further inorganic ions and the formation of a layered precipitate structure [15], but the effect of these ions is currently encapsulated in the reaction rates and the diffusion coefficients of the developed mathematical model. The summary of the considered chemistry to develop the mathematical model is depicted in Fig. 3.1.

### 3.3.2 Mathematical modeling

To keep track of the concentration changes of various contributing chemical components, we define four state variables for the concentration of Mg<sup>2+</sup> ions, protective film (Mg(OH)<sub>2</sub>), chloride (Cl<sup>-</sup>) ions, and the hydroxide (OH<sup>-</sup>)

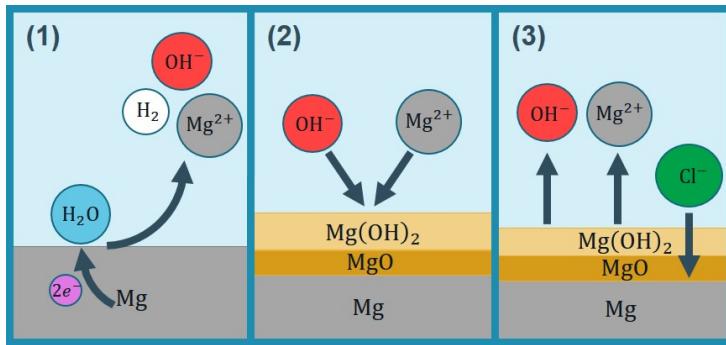


Figure 3.1: The chemistry of biodegradation of Mg considered in the current study: 1) Mg oxidation and water reduction processes accompanied by releasing  $Mg^{2+}$  and  $OH^-$  ions as well as  $H_2$  gas, 2) formation of a partially protective precipitation layer, 3) dynamic solubility equilibrium and contribution of  $Cl^-$ .

ions:

$$\begin{aligned} C_{Mg} &= C_{Mg}(\mathbf{x}, t), \quad C_{Film} = C_{Film}(\mathbf{x}, t) \\ C_{Cl} &= C_{Cl}(\mathbf{x}, t), \quad C_{OH} = C_{OH}(\mathbf{x}, t) \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3' \end{aligned} \quad (3.8)$$

which are indeed 4 scalar functions of space and time.  $\Omega$  denotes the whole region of interest, including both the Mg bulk and its surrounding medium. By doing this, the value of pH at each point of  $\Omega$  can be calculated as:

$$pH = 14 + \log_{10} C_{OH}, \quad (3.9)$$

where  $C_{OH}$  implies the activity of  $OH^-$ . By having the definition of the state variables in Eq. 3.8, the biodegradation of Mg described by Eqs. 3.5 and 3.6 can be represented as a set of reaction-diffusion PDEs:

$$\frac{\partial C_{Mg}}{\partial t} = \nabla \cdot (D_{Mg}^e \nabla C_{Mg}) - k_1 C_{Mg} \left( 1 - \beta \frac{C_{Film}}{[Film]_{max}} \right) + k_2 C_{Film} C_{Cl}^2 \quad (3.10)$$

$$\frac{\partial C_{Film}}{\partial t} = k_1 C_{Mg} \left( 1 - \beta \frac{C_{Film}}{[Film]_{max}} \right) - k_2 C_{Film} C_{Cl}^2 \quad (3.11)$$

$$\frac{\partial C_{Cl}}{\partial t} = \nabla \cdot (D_{Cl}^e \nabla C_{Cl}) \quad (3.12)$$

$$\frac{\partial C_{\text{OH}}}{\partial t} = \nabla \cdot (D_{\text{OH}}^e \nabla C_{\text{OH}}) + k_2 C_{\text{Film}} C_{\text{Cl}}^2 \quad (3.13)$$

in which the maximum concentration of the protective film can be calculated according to its porosity ( $\epsilon$ ) [49]:

$$[\text{Film}]_{\text{max}} = \rho_{\text{Mg(OH)}_2} \times (1 - \epsilon). \quad (3.14)$$

$D^e$  is the effective diffusion coefficient for each component. Due to the formation of the protective film, the diffusion coefficient is not constant and varies from the actual diffusion coefficient of the ions to a certain fraction of it. This fraction can be defined as  $\epsilon/\tau$  [57, 58], in which  $\epsilon$  and  $\tau$  are the porosity and tortuosity of the protective film, respectively. The effective diffusion coefficient can be then calculated by interpolating the two aforementioned values:

$$D_i^e = D_i \left( \left( 1 - \beta \frac{C_{\text{Film}}}{[\text{Film}]_{\text{max}}} \right) + \beta \frac{C_{\text{Film}}}{[\text{Film}]_{\text{max}}} \frac{\epsilon}{\tau} \right). \quad (3.15)$$

The  $\beta$  coefficient is called momentum here and controls the effect of the saturation term  $(1 - \frac{C_{\text{Film}}}{[\text{Film}]_{\text{max}}})$ . The derivation of these equations is discussed in detail in our previous work [59].

### 3.3.3 Interface movement formulation

In order to take advantage of the level set method for tracking the corrosion front, the velocity of the interface at each point should be determined. Then, by solving Eq. 3.4, the interface is obtained at the points with a zero value of the  $\phi$  function. The interface velocity in mass transfer problems can be calculated using the RankineHugoniot equation [60], and by considering the transportation of  $\text{Mg}^{2+}$  ions, it can be written as:

$$\{ \mathbf{J}(x, t) - ([\text{Mg}]_{\text{sol}} - [\text{Mg}]_{\text{sat}}) \mathbf{V}(x, t) \} \cdot n = 0 \quad (3.16)$$

where  $\mathbf{J}$  is the mass flux at the interface. Rearranging Eq. 3.16 and inserting the value of the normal interface velocity into Eq. 3.4 yields:

$$\frac{\partial \phi}{\partial t} - \frac{D_{\text{Mg}}^e \nabla_n C_{\text{Mg}}}{[\text{Mg}]_{\text{sol}} - [\text{Mg}]_{\text{sat}}} |\nabla \phi| = 0, \quad (3.17)$$

which is the final form of the level set equation to be solved. In the case of simulations with a low diffusion rate, the interface moves slowly in the

beginning, which results in a linear degradation, whereas based on the experimental results, the degradation rate is fast at the beginning and slows down eventually [16]. So, to mimic the same behavior in the low diffusion regimes, we took advantage of the theoretical Stefan formulation (Eqs. 3.1 and 3.2) to push the interface in the first time step. According to Eq. 3.1, the velocity of the interface can be calculated as  $(2\alpha/\sqrt{t})$ , but as we are dealing with a 3D model and not a 1D one, we pick a fraction (denoted by  $\gamma$ ) of this ideal value to be used as the driving force of the interface at the beginning of the simulation. So, the normal velocity of the interface can be written in the general form as:

$$V^N(x, t) = \begin{cases} \gamma \frac{2\alpha}{\sqrt{t}} & t = 0 \\ \frac{D_{Mg}^e \nabla_n C_{Mg}}{[Mg]_{sol} - [Mg]_{sat}} & t > 0 \end{cases} \quad (3.18)$$

in which the  $\alpha$  value should be calculated from Eq. 3.2. By selecting  $\gamma$  equal to zero, the Stefan formulation can be eliminated, and a value of 1 for  $\gamma$  restores the ideal 1D velocity definition.

### 3.3.4 Boundary conditions

The implementation of boundary conditions is relatively challenging and complex for the developed model as they should be imposed inside the domain of interest on virtual interfaces defined by mathematical expressions (i.e. on the moving interface defined by the zero iso-contour of the level set equation). The penalty method was used to overcome this issue and define the desired boundary conditions on the moving corrosion front.

Fig. 3.2 demonstrates a schematic presentation of the boundary conditions and general considerations of each PDE of the biodegradation mathematical model. This figure is divided into 5 different parts, presenting the 5 PDEs of the model. The Mg block is depicted in the center, and the interface separates it from the surrounding medium. There is no specific boundary condition for the level set and film formation equations, but in comparison to the other 3 transport equations, it should be noted that diffusivity is not considered for  $Mg(OH)_2$ , which is also reflected in Eq. 3.11. The level set function  $\phi$  is defined in a way that is positive inside and negative outside the solid region. For the  $Mg^{2+}$  ions transport equation, a Dirichlet boundary condition is applied on the mathematical interface to make the concentration equal to the saturation concentration of  $Mg^{2+}$  ions, a value that was already used in Eq. 3.17. For

the  $\text{Cl}^-$  and  $\text{OH}^-$  ions transport equations, a no-flux boundary condition is applied to the interface by making the diffusion coefficient equal to zero inside the Mg block, preventing ions to diffuse inside the solid material.

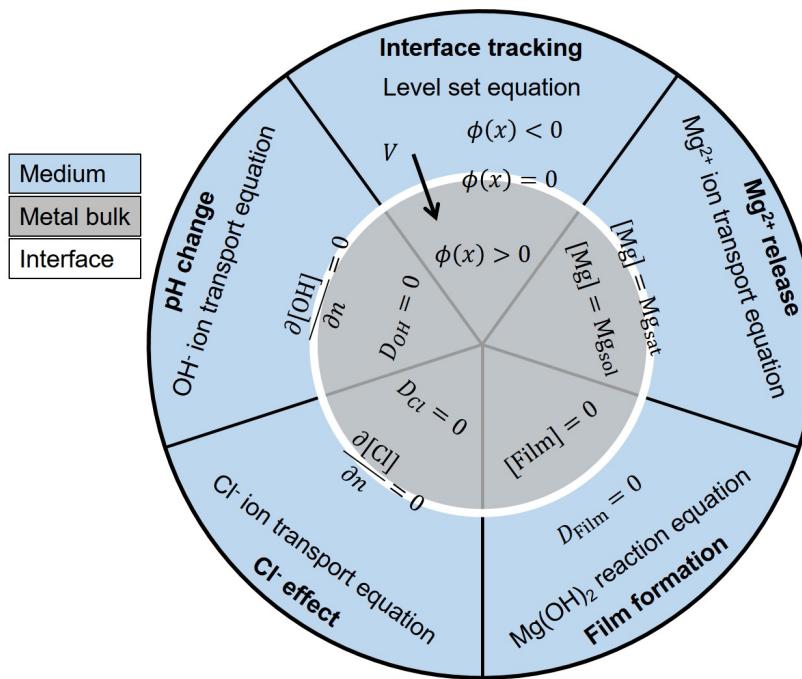


Figure 3.2: A schematic overview of the exposed boundary conditions and constraints required for the simulation of each equation of the developed mathematical model for Mg biodegradation.

### 3.3.5 Implementation

To simulate the developed mathematical model, which is comprised of Eqs. 3.10, 3.11, 3.12, 3.13, and 3.17, a combination of finite difference and finite element methods was used, leading to discrete forms of these equations, which were subsequently solved using appropriate linear solvers.

To discretize the temporal terms of the aforementioned parabolic PDEs, a first-order backward Euler finite difference scheme was used, whereas the spatial terms were converted to a weak form using a standard first-order finite

element scheme. Then, the open-source PDE solver FreeFEM [61] was used to implement the weak form and obtain a linear system of equations for each PDE. The obtained linear systems were solved in parallel using the HYPRE preconditioner [62] and the GMRES solver [63] via the open-source high-performance computing (HPC) toolkit PETSc [64]. Additionally, to increase the efficiency of the computation and decrease the simulation execution time, the computational mesh was decomposed and distributed among available computing resources using the interface of HPDDM package in FreeFEM [65]. The details of this implementation are presented in our previous work [59] as well as in the supplementary materials of this paper. A simple iterative solver based on the Newton method was also developed to solve Eq. 3.2 to obtain the value of  $\alpha$  parameter if it was required in the simulations.

The computational mesh was generated using a set of first-order tetrahedral elements and was adaptively refined on the metal-solution interface to increase the numerical accuracy of the simulation of the level set equation (Eq. 3.17). The Netgen mesh engine [66] in the SALOME platform [67] was used to generate the mesh.

Similar to the technique employed by Bajger et al. [49], the gradient of concentration of  $Mg^{2+}$  in Eq. 3.17 was calculated at a distance  $h$  in the normal direction from the interface, with  $h$  being the smallest element size of the mesh:

$$\nabla_n C = \frac{C(\mathbf{x} + h\mathbf{n}) - C(\mathbf{x} + 2h\mathbf{n})}{h} \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3. \quad (3.19)$$

Considering the adaptively refined mesh, the  $h$  value is very small, so the gradient is computed at the regions close enough to the interface. In addition to this technique, the mass lumping feature of FreeFEM was used to prevent the oscillation of concentration values on the diffusive metal-medium interface.

### 3.3.6 Experimental setup

The degradation rate of CP Mg was evaluated based on the hydrogen evolution tests performed either in NaCl or SBF solutions with eudiometers. The composition of the electrolytes is shown in the following table (Table 3.1). 0.5 g metallic chips (with a surface area of  $47.7 \pm 5.0 \text{ cm}^2/\text{g}$  and chip thickness ca. 200 microns) of CP Mg were put in 500 ml electrolyte for 22-24 hours for monitoring the amount of evolved hydrogen. The method of measuring

evolved hydrogen was chosen for monitoring the degradation rate because although such a measurement is prone to experimental errors such as relatively high solubility of hydrogen in water and volume change due to temperature and pressure variations, it provides a continuous assessment of the process, resulting in a continuous and smooth curve. Additionally, as small metallic chips were used for the tests, it was not possible to clean these pieces in chromic acid without losing them to measure the mass loss directly. The drawback of choosing the evolved hydrogen as the monitoring method is that it is not the only occurring reaction since oxygen reduction also takes place during the process [37, 56, 32]. As a result, measuring only hydrogen does not capture the totality of the degradation reactions. However, for CP Mg, the contribution of oxygen reduction is low (in contrast to high-purity Mg [37]) and can simply be ignored, meaning that the evolved hydrogen is an accurate equivalent for the mass loss. The bulk pH of electrolytes before and after corrosion was measured by a pH meter (Metrohm-691, Switzerland). Local pH was measured by positioning pH microprobes (Unisense, Denmark, pH-sensitive tip size 10x50 micron) 50 micron above the surface of Mg and monitoring the pH values either in one spot or by horizontal or vertical line-scans or mapping by following a horizontal grid. The electrolytes were not pH buffered additionally since SBF contains carbonates and phosphates that stabilize the pH at the approximate value of 8.5 instead of the 10.5 characteristic for pure NaCl solutions where pH is stabilized by precipitation of  $Mg(OH)_2$ . Meanwhile, synthetic pH buffers, such as TRIS and HEPES were proven to affect the degradation mechanism rather significantly and should not be used for this purpose [15]. The measurements were performed at room temperature of  $22 \pm 2^\circ C$  maintained by the laboratory climate control system. More detailed information about experimental set up and procedures can be found elsewhere [16, 20].

### 3.3.7 Parameter estimation

The constructed mathematical model contains some parameters that need to be calibrated prior to final validation of the model: diffusion coefficient of  $Mg^{2+}$  and  $Cl^-$  ions ( $D_{Mg}$  and  $D_{Cl}$  to be inserted into Eq. 3.15 to get effective diffusion coefficients), the reaction rates of Eqs. 3.5 and 3.6 ( $k_1$  and  $k_2$ ), the momentum parameter,  $\beta$ , for controlling the saturation term behavior (in Eqs. 3.10, 3.11, and 3.15), and the  $\gamma$  parameter for the initial interface velocity (Eq.

Table 3.1: Chemical composition of NaCl and SBF electrolytes used to perform hydrogen evolution tests, weight loss, local and bulk pH measurements.

	Concentration/ mM	
	0.85 wt. % NaCl	SBF
Na <sup>+</sup>	145.4	142.0
K <sup>+</sup>	-	5.0
Mg <sup>2+</sup>	-	1.5
Ca <sup>2+</sup>	-	2.5
Cl <sup>-</sup>	145.4	147.8
HCO <sub>3</sub> <sup>-</sup>	-	4.2
HPO <sub>4</sub> <sup>2-</sup> / H <sub>2</sub> PO <sub>4</sub> <sup>-</sup>	-	1.0
SO <sub>4</sub> <sup>2-</sup>	-	4.2
Synthetic pH buffer (i.e. Tris/HCl, HEPES)	No	No
Initial pH value	5.6-5.9	7.35-7.45

3.18). An inverse problem setup was required to estimate the proper value of these parameters.

Performing a parameter estimation requires running the computational models several times. Considering the computationally-intensive model of the current study, a sensitivity analysis was performed prior to the parameter estimation to exclude non-essential parameters and reduce the time required to complete the inverse problem run. This sensitivity analysis was accomplished separately in the low diffusion (similar to the SBF solution) and high diffusion (similar to NaCl solution) regimes.

After determining the essential parameters to include, a Bayesian optimization approach [68] was used to construct the inverse problem and calibrate the parameters. The reason for choosing a Bayesian approach was to minimize the number of optimization iterations, in each of which the simulation should run once. The Bayesian optimization is a more efficient option for such computational intensive cases in comparison to gradient-based or fully-stochastic methods as it takes into account all the preceding iterations in a probability tree [69].

The objective function of the optimization problem was the difference between the predicted and experimentally obtained values of evolved hydrogen. In the computational model, the evolved hydrogen can be computed directly at any

time through the mass loss as each mole of corroded Mg is correlated to one mole of released hydrogen (Eq. 3.5). The mass loss can be obtained using the following volume integral:

$$\text{Mg}_{\text{lost}} = \int_{\Omega_+(t)} [\text{Mg}]_{\text{sol}} dV - \int_{\Omega_+(0)} [\text{Mg}]_{\text{sol}} dV, \quad (3.20)$$

where  $\Omega_+(t) = \{\mathbf{x} : \phi(\mathbf{x}, t) \geq 0\}$ , and then, the amount of produced hydrogen is calculated using the ideal gas law:

$$H_f = \frac{\text{Mg}_{\text{lost}}}{\text{Mg}_{\text{mol}}} \frac{RT}{P} \quad (3.21)$$

in which  $R$ ,  $P$ ,  $T$ ,  $\text{Mg}_{\text{mol}}$  are the universal gas constant, the pressure, the medium temperature, and the molar mass of Mg, respectively.

### 3.3.8 Simulation setup

In order to simulate the developed mathematical model, the experimental setup was reconstructed *in silico* with some minor differences. As there is no perfusion in the solution chamber, the mixing effect was neglected, so, as can also be seen in the mathematical model, the advection terms were not considered. Furthermore, the experiments were conducted using small metallic chips, yet, as the biodegradation behavior heavily depends on the exposed surfaces, we represented these chips by a cuboid with the same surface-to-mass ratio. By considering the approximate surface-to-mass of  $50\text{cm}^2/\text{g}$  and the total mass of  $0.5\text{g}$ , the chips were replaced by a cuboid with the size of  $60\text{mm} \times 21\text{mm} \times 0.2\text{mm}$ , which approximately has the same ratio, surface area, volume, and mass. Also, the solution chamber with a capacity of  $500\text{ml}$  was represented by a cubic container with an edge size of  $80\text{mm}$ . Fig. 3.3 depicts the constructed geometry as well as the computational mesh generated to represent it. The mesh is refined on the interface and contains 18,049,471 elements, resulting in 3,077,227 degrees of freedom (DOF) for each PDE.

Simulations were carried out on the VSC (Flemish Supercomputer Center) supercomputer. Taking advantage of HPC techniques to parallelize the simulation is an inevitable aspect of such a computational-intensive model, so based on what described in the implementation section, the mesh was decomposed among 170 computing cores, i.e. 24,137 DOF per core (which includes the

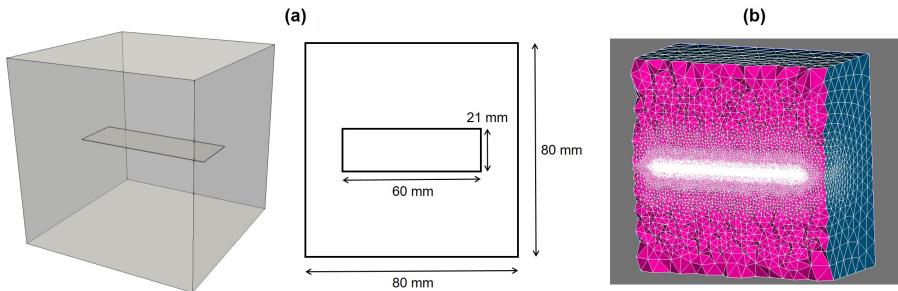


Figure 3.3: Computational representation of the experimental set-up, used to perform parameter estimation and numerical validation of the developed model. a) A cuboid of Mg ( $60\text{mm} \times 21\text{mm} \times 0.2\text{mm}$ ) inside a solution, b) a cross-section of the computational mesh, refined on the corrosion front to increase the required level set accuracy.

ghost nodes to satisfy the boundary condition in each sub-mesh). On the VSC supercomputer, we made use of 5 nodes, 36 cores each, each node holding CPUs with a clock speed of 2.6 GHz, with 960 GB of the total available memory.

The  $\text{OH}^-$  transport equation (Eq. 3.13) was not solved during the parameter calibration process. Afterwards, two full simulations (for the NaCl and SBF solutions) were conducted to calculate the pH changes based on the change of the concentration of  $\text{OH}^-$  ions in the medium. This acted as the validation of the numerical model because no calibration was performed on the output of this equation. The pH was calculated using Eq. 3.9, based on the solution of Eq. 3.13 and a reported value of  $7.00e \times 10^{-5}\text{cm}^2/\text{s}$  ( $25.2\text{mm}^2/\text{hour}$ ) for the diffusion coefficient of  $\text{OH}^-$  ions ( $D_{\text{OH}}$  to be used in Eq. 3.15) in aqueous solutions [70].

According to the experimental setup, the initial concentration of the  $\text{Mg}^{2+}$ ,  $\text{Cl}^-$ , and  $\text{OH}^-$  ions were set to 0 (no  $\text{Mg}^{2+}$  ions at the beginning),  $146\text{mM}$  ( $5.175 \times 10^{-6}\text{g/mm}^3$ ), and  $1 \times 10^{-7}\text{g/mm}^3$ , respectively. The porosity ( $\epsilon$ ) and tortuosity ( $\tau$ ) of the protective film were considered to be 0.55 and 1, respectively [48]. The saturation concentration  $[\text{Mg}]_{\text{sat}}$  was set to the solubility of magnesium chloride in water, which is  $134 \times 10^{-6}\text{g/mm}^3$  at  $25^\circ\text{C}$  [71]. The density of Mg ( $[\text{Mg}]_{\text{sol}}$ ) and  $\text{Mg}(\text{OH})_2$  were set to  $1735 \times 10^{-6}\text{g/mm}^3$  and  $2344 \times 10^{-6}\text{g/mm}^3$ , respectively [49]. A time step convergence study was performed to determine the implicit time step size. Based on the results, a time

step with a size of 0.025 hours was chosen. The overall simulated time is 22 hours in accordance with the experimental design of performed immersion tests.

### 3.3.9 Case study

To further investigate the predictions of the current model on more complex shapes, the biodegradation of a simple screw was studied in the SBF solution using the parameters obtained for the low diffusion regimes. Similar to the simulation of Mg cuboid, the mesh was refined on the metal-medium interface, and it consisted of 1,440,439 elements with 246,580 DOFs for each PDE. All the simulation parameters and materials properties were identical to the simulation of biodegradation in the SBF solution, and the target was to simulate 42 days (1008 hours) of the process. This was selected as a sufficiently long simulated time to observe the effects of biodegradation on larger time scales.

## 3.4 Results

### 3.4.1 Optimization results

Based on the performed sensitivity analysis, two parameter sets were obtained for the high diffusion (in NaCl solution) and low diffusion (in SBF solution) simulations, respectively. These parameters are listed in Table 3.2. According to the results, the reaction rate of Eq. 3.5 ( $k_1$ ), which demonstrates the rate of oxidation-reduction, has less contribution to the process in comparison to the rate of the weakening of the protective film ( $k_2$ ). Because of this, the parameter  $k_1$  was not selected for the parameter estimation. Also, the model was sensitive to the effect of parameter  $k_2$  in different ranges of values and not on a specific point, and as a result, three constant values were chosen as the delegates of these ranges in the optimization process. The model was not sensitive to the diffusion rate of  $\text{Cl}^-$  ions, which was also expected because although  $\text{Cl}^-$  has an important role in the weakening of the partially protective  $\text{MgO}$  film, its transport equation (Eq. 3.12) is purely diffusive and does not include any reaction term.

Table 3.2: The effective parameters as the result of the sensitivity analysis and their corresponding range to be considered in the Bayesian optimization for parameter calibration

	Parameter	Optimization range
Low diffusion (SBF solution)	$D_{\text{Mg}}$	[0.0001, 0.01]
	$k_2$	$10^{10}, 10^{15}, 10^{20}$
	$\beta$	[0.1, 10]
High diffusion (NaCl solution)	$\gamma$	[0, 1]
	$D_{\text{Mg}}$	[0.003, 0.1]
	$k_2$	$10^{10}, 10^{15}, 10^{20}$
	$\beta$	[0.1, 10]

Table 3.3: Values used to evaluate the model performance, obtained from the output of the optimization process and the literature.

Parameter	$D_{\text{Mg}}$	$D_{\text{Cl}}$	$D_{\text{OH}}$	$k_1$	$k_2$	$\beta$	$\gamma$
Unit	$\frac{\text{mm}^2}{\text{hour}}$	$\frac{\text{mm}^2}{\text{hour}}$	$\frac{\text{mm}^2}{\text{hour}}$	$\frac{1}{\text{hour}}$	$\frac{\text{mm}^6}{\text{g}^2\text{hour}}$	-	-
SBF solution	0.000338	0.05	25.2	7	$10^{15}$	0.125	0.65
NaCl solution	0.06273	0.05	25.2	7	$10^{20}$	0.2	0

The parameter optimization process was performed on the specified range of selected parameters, while the rest of the parameter values were obtained from the literature [49, 70]. Table 3.3 shows the output of this process, which was used to simulate the full model. For two estimation processes, 120 optimization iterations were taken cumulatively, which took 276 hours of simulation execution time using 170 computing nodes for each simulation.

### 3.4.2 Degradation prediction

Fig. 3.4 shows the model output for the predicted produced hydrogen, protective film formation, and the pH changes. The graph of the evolved hydrogen is used as input during the parameter optimization process, but the pH results are produced by the simulations using the optimized parameters to demonstrate the validation of the developed mathematical and computational models. The predicted pH result (Fig. 3.4-d) shows a difference of 5.35% for the simulation in NaCl and 1.03% for SBF simulation. Each simulation took about 3 hours to complete.

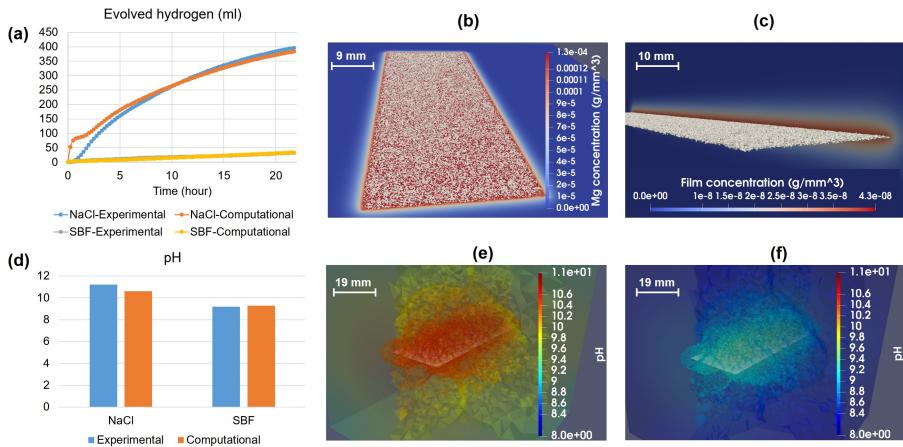


Figure 3.4: Comparing the quantitative output of the model for the rate of degradation and the pH changes in NaCl and SBF solutions with experimentally measured values as well as the simulation results for ion release, mass loss, protective film formation, and pH changes after 22 hours of simulated time: a) calibrated output of the formed hydrogen gas during the degradation (the SBF curves are overlapped), b) the simulation results of  $Mg^{2+}$  ions release, c) the simulation results of protective film concentration at the end of the simulation (the color contour shows the concentration of species, and the gray surface is the zero iso-contour of the level set function, which indicates the surface of the Mg block), d) de novo prediction of the global pH changes in the medium, showing a good agreement between the model output and the experimental results, e) pH changes in different regions of the medium in NaCl solution, f) pH changes in SBF solution.

In Fig. 3.4, a post-processed view of the final shape of the Mg cuboid in the NaCl solution is presented, in which the degraded geometry is plotted on the  $Mg^{2+}$  ions (Fig. 3.4-b) and protective film concentration (Fig. 3.4-c) contours. A transparent contour of the pH values in the solution is depicted for both the NaCl (Fig. 3.4-e) and SBF (Fig. 3.4-f) solutions. The range of colors is kept equal for both contours to make it easy to compare the change of pH in both solutions.

The concentration values of the state variables of the derived transport PDEs ( $Mg^{2+}$ ,  $Cl^-$ ,  $OH^-$ , and  $Mg(OH)_2$ ) are plotted along a diagonal line in the solution container in Fig. 3.5, showing how they change in the zones close to

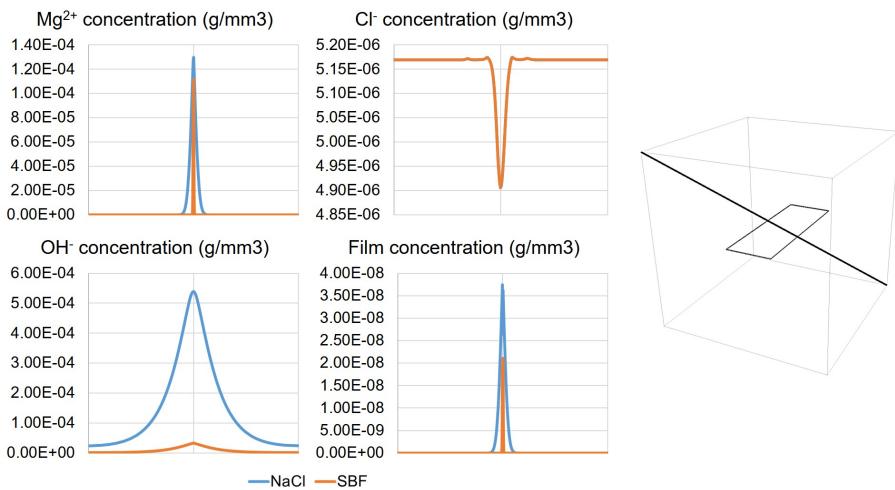


Figure 3.5: The change of concentration for the involved chemical components,  $\text{Mg}^{2+}$ ,  $\text{Cl}^-$ ,  $\text{OH}^-$  ions, and the protective precipitation structure (which can be correlated to the thickness of the layer) plotted over a diagonal line as shown in the right.

the corrosion surface and far from it. The animated output of the degradation of the Mg cuboid is presented as a set of movie files in the supplementary materials of this paper (post-processed using NVIDIA IndeX software on a GPU).

### 3.4.3 Example application

The simulation of 42 days (19,200 time steps) of the degradation of the simple screw took 9 hours to run using 170 computing cores. Fig. 3.6 depicts the post-processed interface and  $\text{Mg}^{2+}$  ions release (similar to Fig. 3.4-b) as well as the mass loss during the degradation of the screw in the SBF solution.

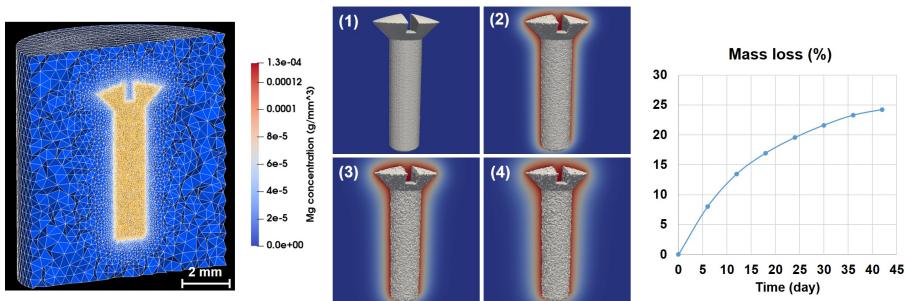


Figure 3.6: A cross-section of the computational mesh and simulation results of the degradation process of the use-case screw in SBF solution as well as the mass loss graph over time. The contours display the concentration of  $Mg^{2+}$  ions on a cross-section view of the medium beside the moving surface of the screw at 1) 1st day (initial state), 2) 6th day, 3) 12th day, and 4) 18th day.

### 3.5 Discussion

In this study, a physicochemical model of the biodegradation process of commercially-pure Mg was developed by constructing a mathematical model formulating the mass transfer phenomena as well as tracking the location of the surface of the implant during degradation. For the mass transfer model, the equations were derived from the chemistry of biodegradation of the Mg in saline (NaCl) and buffered (SBF) solutions, which includes the oxidation of the metallic part, reduction of water, changes in pH, and formation of a protective film on the surface of the scaffold which contributes to a slower rate of degradation. Beside these aspects, it was also crucial to consider the effect of different ions in the medium on the rate of degradation. Additionally, investigating the structural changes of the scaffolds and implants in practical applications, like resorption of temporary fixation devices, requires tracking the movement of the corrosion surface. This was done by constructing an equation based on the level set principle, which captured the movement of the medium-metal interface by defining an implicit surface. The derived equations were coupled and solved using a combination of finite difference and finite element methods. The degradation data to validate the models was collected from immersion tests of small Mg chips, reconstructed as a single cuboid in the computational study with a similar surface over volume properties. The model parameters were calibrated using a Bayesian optimization algorithm, and the

obtained parameters were used to simulate the pH changes in NaCl and SBF solutions.

The developed model falls in the categories of physical models of the corrosion process, which provide more insights of the process in comparison to the phenomenological models. The reason is that the phenomenological models focus on the elimination of elements to capture the loss of materials, which makes it impossible to model the formation of new chemical compounds or interaction of species [72]. The physical models, like the one developed in this study, are capable of capturing the underlying chemical interactions. By doing this, processes like the effect of coating, the formation of a protective layer, and pH changes can be modeled. Adding an appropriate interface tracking method enables the physical models to act like the phenomenological models in capturing the corrosion interface movement. In the current study, this has been accomplished using a level set function. Technically speaking, this approach has certain benefits over the ALE method, which is the method used by several similar studies, including Grogan et al. [44]. In comparison to the ALE method, the level set function tracks the interface instead of a Lagrangian mesh, and elements can freely be marked as solid or liquid. Additionally, employing the ALE method for degradation simulation requires remeshing the geometry as the interface moves, which is not efficient for 3D models and is limited to the available features of the employed numerical solver.

One of the challenging aspects of validating physical models is getting the correct value for the parameters of said models, requiring dedicated experimental input. To overcome this challenge, an efficient inverse problem was constructed based on the Bayesian optimization approach to estimate the unknown parameters. To save time and resources, the parameter estimation process was performed on the most effective parameters, which were selected based on a sensitivity analysis. This selection process implied the importance of parameters in high and low diffusion rates.

The degradation rate is fast at the beginning, but then it slows down due to the formation of a partially protective film and also because of the saturation concentration. This phenomenon is well captured by the model at high diffusion rates, but in low diffusion rates (in SBF solution), this effect can be reproduced by pushing the corrosion front according to the Stefan formulation of the moving interface problems. This was controlled by the parameter  $\gamma$  (Eq. 3.18). It should be noted that the inclusion of the  $\gamma$  parameter is crucial for short-term simulations only, helping the model mimic the chemical

behavior correctly. Defining and considering  $\gamma$  is necessary because from the computational costs perspective, performing the parameter calibration on simulations with thousands of time steps requires a lot more resources and time. For the high diffusion regime simulation, the results show a difference between the experimental and computational data in the early stages of the degradation process (Fig. 3.4-a). The reason for this behavior lies within Eqs. 3.17 and 3.19, in which the interface velocity was correlated to the gradient of released ions. In high diffusion rates, the material release occurs very fast, so the calculated gradient (Eq. 3.19) vanishes for a short period until the diffusion becomes more uniform. As a result, the interface does not move, and according to Eq. 3.20, no mass loss gets calculated. This effect was automatically ignored in the parameter estimation process since the objective function considers the overall degradation behavior.

The degradation of the CP Mg was assumed to be mostly diffusion-based. As a result, the value of  $D_{\text{Mg}}$  plays an important role in the behavior of the model. Although it was possible to get the diffusion coefficient of  $\text{Mg}^{2+}$  from the previously conducted experiments in the literature (similar to what was done for  $D_{\text{OH}}$ ), we decided to not do so because of two reasons: 1) the values reported in the literature are mostly valid for saline solutions only, and 2) the reported values were not in a good agreement with one another [44, 48]. Thus, the diffusion coefficient was obtained using the parameter estimation process for both the NaCl and SBF solutions. The obtained value of  $D_{\text{Mg}}$  ( $0.06273 \text{ mm}^2/\text{hour}$ ) was in line with the values that Grogan et al. have already suggested ( $0.010575 - 0.50575 \text{ mm}^2/\text{hour}$ ) [44], showing that the constructed inverse problem was successful in reproducing previous results of similar studies. The obtained value of  $D_{\text{Mg}}$  in the Bajger et al. work [49] is  $0.00066 \text{ mm}^2/\text{hour}$ , which is mostly related to the simplicity of the employed parameter estimation method as well as having a 2D model instead of a 3D one.

In the *in vitro* biodegradation of Mg-based biomaterials, the local pH of the surrounding solution increases less than that in NaCl solution. This is because the  $\text{Mg}(\text{OH})_2$  formed in NaCl stabilizes pH at 10.4 [73], while Mg-Ca-P-C containing products stabilize the pH at 7.8-8.5 since  $\text{OH}^-$  is consumed for the formation of this product [19, 16]. This phenomenon was captured in Eqs. 3.13 and 3.9 to calculate pH based on the concentration of  $\text{OH}^-$  ions, showing the local pH changes at any location (Fig. 3.4-e,f). In the current study, the global pH is considered as the validation criterion, which means that the average

value of the solution pH is calculated using a volume integral and is compared with the ones obtained from the experiments. Fig. 3.4-d shows that such a prediction has a good agreement with the experimental data.

One of the biggest simplifications of the current study was made by ignoring the contribution of pH changes to the biodegradation mechanism of Mg. Although doing that is relatively simple and straightforward in the approach taken by this study, it results in non-linear terms in the derived PDEs. This non-linearity inserts another level of complexity to the computational model as the order of the state variables are in the range of  $10^{-6}$  to  $10^{-10}$ , which makes it difficult to yield convergence in the iterative non-linear solvers. By developing a robust non-linear solver, this effect can be added simply by including more relevant terms as the effect of Eq. 3.13 into Eq. 3.10.

Additionally, buffered solutions and the physiological fluids inside the human and animal bodies contain more ions interacting with more complex chemistry [15]. In this study, this effect was encapsulated in a limited number of parameters (such as  $k_1$  and  $k_2$  in Eqs. 3.10, 3.11, and 3.13), but while the results show its success to reproduce experimental observations, it still needs additional elaboration to be able to capture more chemical interactions. For Example, SBF solutions contain phosphates, carbonates, and calcium that form hydroxyapatite-like compounds on the surface of Mg, acting as rather strongly blocking corrosion products. In the current state of the developed model, such an effect on the corrosion rate was captured by a low effective diffusion coefficient (Eq. 3.15) for the Mg transport. In future model developments, the effect of presented inorganic ions such as  $\text{HCO}_3^-$ ,  $\text{HPO}_4^{2-}/\text{H}_2\text{PO}_4^-$ , and  $\text{Ca}^{2+}$  can be added similar to the way the effect of  $\text{Cl}^-$  was considered. Additionally, formulating the effect of  $\text{HPO}_4^{2-}$  that exists in the physiological environments will make the model capable of making more accurate predictions for *in vivo* studies.

A common approach in mechanistic studies is to start with a pure material and gradually increasing complexity by adding impurities and alloying elements. This approach was followed in the current study by beginning with a model for pure Mg that captures the major reactions. The developed model can be further extended to Mg alloys by considering the effect of alloying elements on the reaction rates as well as adding more terms to the transport equations to capture the electrochemical potential changes, converting the PDEs into the NernstPlanck equation [74]. By doing so, more complex forms of the corrosion process, such as galvanic corrosion, can be predicted by the model.

This will increase the applicability of the model for biomedical cases since pure Mg is not commonly used for medical-graded applications. As an additional future development, the corrosion layer can be considered to be heterogeneous, making it possible to simulate the cathodic reactions by randomly distributing more active spots on the surface. Alternatively, a similar effect can be achieved by adapting the degradation rates using polarization curves and introducing an active spot for inhomogeneous anodic dissolution [75]. Applying this will enable the model to take into account additional corrosion products formed due to additional alloying elements such as Zn, Ca, Ag, rare-earth elements, and detrimental cathodic impurities such as Fe.

Although the pH simulations are not enough experimental input to call the model fully validated, the obtained validation results show that the derived mathematical model and the corresponding parallelized computational model give a correct *in silico* representation of the studied process. The performed predictive simulations, including the case study, demonstrate the potential of the developed computational model and software to study the biodegradation behavior of implants. This can be further combined with other computational models to provide a multidisciplinary environment to investigate the mechanical integrity of implants or induced neotissue growth for different applications in orthopedics and tissue engineering.

### 3.6 Conclusions

The use of biodegradable metals for designing medical devices and implants has the challenge of controlling the release and rate of degradation, which is usually investigated by conducting *in vitro* and *in vivo* tests requiring conducting multiple experiments for different scenarios and situations. In this study, we have developed a mathematical model to predict the biodegradation behavior of commercially pure Mg-based biomaterials, which makes it possible to study the corrosion of implants and scaffolds in a simulated environment. Despite the assumed simplifications, the model can serve as an important tool to find the biodegradable metals properties and predict the biodegradation behavior of Mg-based implants that improves current design workflows.

## Acknowledgment

This research is financially supported by the Prosperos project, funded by the Interreg VA Flanders The Netherlands program, CCI grant no. 2014TC16RFCB046 and by the Fund for Scientific Research Flanders (FWO), grant G085018N. LG acknowledges support from the European Research Council under the European Union's Horizon 2020 research and innovation programme, ERC CoG 772418. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government department EWI.

## **Chapter 4**

# **Fluid flow and convection**



# **Chapter 5**

## **Advanced model with kinetics**



# **Chapter 6**

## **Tissue growth**



# Chapter 7

## Model parallelization

This chapter is based on previously published content in *The International Journal of High Performance Computing Applications*:

M. Barzegari, and L. Geris, "Highly scalable numerical simulation of coupled reaction-diffusion systems with moving interfaces," *The International Journal of High Performance Computing Applications*, vol. 36, pp. 198-213, 2022.

### 7.1 Introduction

Moving-boundary problems [54] are a subset of the general concept of boundary-value problems which not only require the solution of the underlying partial differential equation (PDE), but also the determination of the boundary of the domain (or sub-domains) as part of the solution. Moving-boundary problems are usually referred to as Stefan problems [54] and can be used to model a plethora of phenomena ranging from phase separation and multiphase flows in materials engineering to bone development and tumor growth in biology. Reaction-diffusion systems are the mathematical models in which the change of state variables occurs via transformation and spreading. These systems are described by a set of parabolic PDEs and can model a large number of different systems in science and engineering, for instance predator-prey models in biology and chemical components reactions in chemistry [53]. Combining the reaction-diffusion systems with moving-boundary problems

provides a way to study the systems in which the diffusion and reaction lead to the change of domain geometry. Such systems have great importance in various real-world scenarios in chemistry and chemical engineering as well as environmental and life sciences.

In this study, the material degradation phenomenon has been investigated as an example of a reaction-diffusion system with moving boundaries, in which the loss of material due to corrosion leads to movement of the interface of the bulk material and surrounding corrosion environment. More specifically, the degradation of magnesium (Mg) in simulated body fluid has been chosen as a case study. Magnesium has been chosen due to its growing usability as a degradable material in biomedicine, where it is usually used in biodegradable implants for bone tissue engineering and cardiovascular applications [38, 4]. The ultimate application of such a model can be then to study the degradation behavior of resorbable Mg-based biomaterials.

A wide range of different techniques has already been developed to study the moving interfaces in reaction-diffusion problems, which can be grouped into 3 main categories: 1) mesh elimination techniques, in which some elements are eliminated to simulate the interface movement (or loss of material in corrosion problems), 2) explicit surface representation, such as the arbitrary Lagrangian-Eulerian (ALE) method, which tracks the interface by moving a Lagrangian mesh inside an Eulerian grid, and 3) implicit surface tracking, in which an implicit criterion is responsible to define the moving interface during the reaction-diffusion process. Related to the aforementioned case study, studies performed by [51] and [42] are examples of the first group. [51] have constructed a simulation of degradation using the mesh elimination technique. [42] have developed a continuous damage (CD) model by using an explicit solver to study the degradation. The work of [44] is an example of the second group as they have developed one of the first models to correlate the mass flux of the metallic ions in the biodegradation interface to the velocity of said interface. This was used to build an ALE model to explicitly track the boundary of the material during degradation. Studies of the third category are based more on mathematical modeling rather than available models in simulation software packages. This approach results in more flexibility and control over the implementation of the computational model. For instance, [46] have derived a system of mathematical equations to study galvanic corrosion of metals, taking advantage of the level set method (LSM) to track the corrosion front. [49] have used the definition of velocity of the biodegradation interface

as the speed of the moving boundary in LSM, enabling them to track the geometrical changes of the material during degradation. Similarly, [76] have used a combination of LSM and extended finite element method (XFEM), a method to model regions with spatial discontinuities, to study the moving corrosion front in the pitting corrosion process. A very similar approach and formulation has been taken by [77] to model localized pitting corrosion. An alternative method for tracking the moving interface is the phase field method, which has been used in a wide range of relevant studies. A comparison between the behavior of phase-field and LSM formulations for an evolving solid-liquid interface has been performed by [78], showing that both methods lead to the same results for diffusion-reaction systems. The approach taken in this study was similar to the one from Bjger et al., where LSM was employed to correlate the diffusion and reaction processes to the movement of the solid-solution interface using continuous variables.

Tracking the moving front at the diffusion interface requires high numerical accuracy of the diffusive state variables, which can be achieved using a refined computational grid. This makes the model computationally intensive, and as a consequence, implementing parallelization is an inevitable aspect of simulating such a model. Such an approach enables the model to simulate large-scale systems with a large number of degrees of freedom (DOF) in 3D with higher performance and efficiency in high-performance computing (HPC) environments. In recent years, parallelization of diffusion-reaction systems simulation has been investigated, but the studies are mainly conducted for stochastic (statistical) models. For instance, [79] have developed a parallel stochastic model for large-scale spatial reaction-diffusion simulation, and similarly, [80] have developed a stochastic high-performance simulator for specific biological applications. Also as an example for massively parallel systems, [81] have conducted a simulation of reaction-diffusion processes in biology using graphics processing units (GPUs). Although stochastic models have more parallel-friendly algorithms, explaining the underlying process, especially when it involves reaction-diffusion processes of chemistry and biology, is less complex and more universal using mechanistic (deterministic) models, which are based on well-developed mathematical models of continuous systems [82]. To the best of authors' knowledge, none of the previous contributions to the topic of reaction-diffusion systems with moving interfaces has employed parallelization techniques to increase the performance and speed of execution of the model without compromising the accuracy of the interface tracking.

In the current study, we developed a mechanistic model of a reaction-diffusion system coupled with a moving interface problem. Improving the accuracy of the interface capturing requires a refined computational mesh, leading to a more computation-intensive simulation. To overcome this challenge and yield more interactable simulations, scalable parallelization techniques were implemented making the model capable of being run on massively parallel systems to reduce the simulation time. The investigated case-study is the material degradation process. The developed model captures the release of metallic ions to the medium, formation of a protective film on the surface of the material, the effect of presented ions in the medium on the thickness of this protection layer, and tracking of the movement of the corrosion front (Fig. 7.1). The interface tracking was performed using an implicit distance function that defined the position of the interface during degradation. This implicit function was obtained by constructing and solving a level set model. It is also worth noting that in a real-world application, such systems require a calibration (also called parameter estimation or inverse problem), in which the model should be simulated hundreds of times. This makes the parallelization even more crucial for these models.

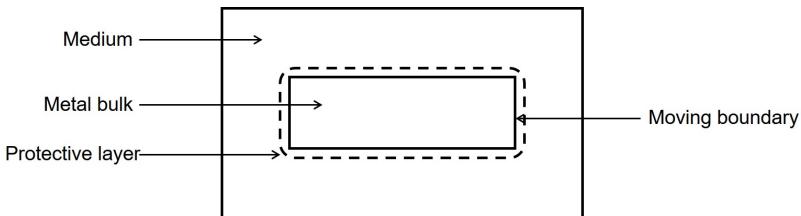


Figure 7.1: A schematic presentation of different components of the developed model for simulation of the degradation process with a moving front.

## 7.2 Background theory and model description

Before elaborating the parallel implementation strategy, the mathematical model is briefly described in this section. The model is constructed based on the chemistry of degradation, starting from the previous work by [49], in which the ions can diffuse to the medium and react with each other.

### 7.2.1 Chemistry of degradation

In metals, degradation occurs through the corrosion process, which usually consists of electrochemical reactions, including anodic and cathodic reactions as well as the formation of side products [1].

For Mg, the corrosion reactions comprise the following steps [1]: first, the material is released as metallic ions and free electrons, which causes the volume of the bulk material to be reduced:



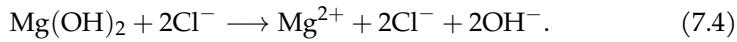
The free electron reduces water to hydrogen gas and hydroxide ions:



Then, with the combination of the metallic and hydroxide ions, a porous film is formed on the surface, slowing down the degradation rate by protecting the material underneath:



With the presence of some specific ions in the surrounding medium, such as chloride ions in a saline solution, the protective film might be broken partially, which contributes to an increase of the rate of degradation:



The degradation process of metals is a continuous repetition of the above reactions.

### 7.2.2 Reaction-diffusion equation

A reaction-diffusion partial differential equation can describe the state of a reaction-diffusion system by tracking the change of the concentration of the different components of the system over time [53]. The equation is a parabolic

PDE and can be expressed as

$$\frac{\partial u}{\partial t} - \nabla \cdot [D \nabla u] = f(u) \quad (7.5)$$

in which the change of the state variable  $u = u(\mathbf{x}, t), \mathbf{x} \in \Omega \subset \mathbb{R}^3$  is described as a combination of how it diffuses and how it is produced or eliminated via reactions. The term  $f(u)$  is a smooth function that describes the reaction processes.

In the example used in this study, the state variable in Eq. 7.5 is the concentration of effective chemical components involved in the degradation process, namely magnesium ions and the protective layer, denoted by  $C_{\text{Mg}}$  and  $C_{\text{Film}}$  respectively.

$$C_{\text{Mg}} = C_{\text{Mg}}(\mathbf{x}, t), \quad C_{\text{Film}} = C_{\text{Film}}(\mathbf{x}, t) \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3 \quad (7.6)$$

$\Omega$  is the whole domain of interest, including the bulk material and its surrounding medium. So, by assuming that the reaction rates of Eqs. 7.3 and 7.4 are  $k_1$  and  $k_2$  respectively, one can write the change of those state variables according to Eq. 7.3 and Eq. 7.4 as

$$\frac{\partial C_{\text{Mg}}}{\partial t} = \nabla \cdot (D_{\text{Mg}}^\epsilon \nabla C_{\text{Mg}}) - k_1 C_{\text{Mg}} + k_2 C_{\text{Film}} [\text{Cl}]^2 \quad (7.7)$$

$$\frac{\partial C_{\text{Film}}}{\partial t} = k_1 C_{\text{Mg}} - k_2 C_{\text{Film}} [\text{Cl}]^2. \quad (7.8)$$

We assumed that the concentration of the chloride ions is constant (denoted by  $[\text{Cl}]$  in the equation) and does not diffuse into the protective film. The missing part of the model described by Eqs. 7.7 and 7.8 is the effect of the protective film on the reduction of the degradation rate. To this end, we defined a saturation term,  $(1 - \frac{C_{\text{Film}}}{[\text{Film}]_{\text{max}}})$  for the concentration of Mg ions in the equations. By considering the film's porosity ( $\epsilon$ ), the maximum concentration of the protective layer can be calculated based on its density ( $\rho_{\text{Mg(OH)}_2}$ ):

$$[\text{Film}]_{\text{max}} = \rho_{\text{Mg(OH)}_2} \cdot (1 - \epsilon). \quad (7.9)$$

The defined saturation term acts as a function of space that varies between 0

and 1 in each point. By adding this term to the concentration of Mg ions, we can write

$$\frac{\partial C_{\text{Mg}}}{\partial t} = \nabla \cdot (D_{\text{Mg}}^e \nabla C_{\text{Mg}}) - k_1 C_{\text{Mg}} \left( 1 - \frac{C_{\text{Film}}}{[\text{Film}]_{\max}} \right) + k_2 C_{\text{Film}} [\text{Cl}]^2 \quad (7.10)$$

$$\frac{\partial C_{\text{Film}}}{\partial t} = k_1 C_{\text{Mg}} \left( 1 - \frac{C_{\text{Film}}}{[\text{Film}]_{\max}} \right) - k_2 C_{\text{Film}} [\text{Cl}]^2. \quad (7.11)$$

Since the film is a porous layer and allows the ions to diffuse through it, the diffusion coefficient in Eq. 7.10 is a function of space and not a constant value (which is the reason for being denoted as  $D_{\text{Mg}}^e$ ). We can calculate this effective diffusion function by interpolating two values at any point: 1)  $D_{\text{Mg}}^e = D_{\text{Mg}}$  when  $C_{\text{Film}} = 0$ , and 2)  $D_{\text{Mg}}^e = \frac{\epsilon}{\tau} D_{\text{Mg}}$  when  $C_{\text{Film}} = [\text{Film}]_{\max}$ , in which  $\epsilon$  and  $\tau$  are the porosity and tortuosity of the protective film, respectively. The interpolation leads to the effective diffusion function:

$$D_{\text{Mg}}^e = D_{\text{Mg}} \left( \left( 1 - \frac{C_{\text{Film}}}{[\text{Film}]_{\max}} \right) + \frac{C_{\text{Film}}}{[\text{Film}]_{\max}} \frac{\epsilon}{\tau} \right). \quad (7.12)$$

### 7.2.3 Level set method

The level set method is a methodology that allows moving interfaces to be described by an implicit function. In other words, the boundaries of domains can be tracked as a function instead of being explicitly defined. In the level set method, a signed distance function,  $\phi = \phi(x, y, z, t)$ , describes the distance of each point in space to the interface, and the zero iso-contour of this function implies the interface [55]. In the current study, this function was defined in a way that divides the domain into two subdomains: 1) the bulk material, in which the implicit function is positive ( $\phi > 0$ ), and 2) the medium, in which the function is negative ( $\phi < 0$ ). The interface is defined as the points in space where  $\phi = 0$ . Fig. 7.2 shows a schematic representation of the solid-medium interface in the current study, in which the interface moves as the material degrades over time.

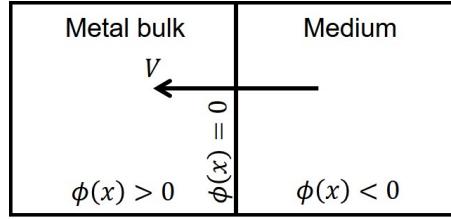


Figure 7.2: A schematic representation of the implicit function definition in the current study.  $V$  denotes the shrinkage speed of the interface due to degradation.

The level set equation defines this implicit function. The full level set equation can be written as [55]:

$$\frac{\partial \phi}{\partial t} + \underbrace{\vec{V^E} \cdot \nabla \phi}_{\text{External velocity field}} + \underbrace{V^N |\nabla \phi|}_{\text{Normal direction motion}} = \underbrace{b\kappa |\nabla \phi|}_{\text{Curvature - dependent term}} \quad (7.13)$$

in which the terms correspond to temporal changes, external velocity field effect, normal direction motion, and curvature-dependent interface movement, respectively.  $\vec{V^E}$  is the external velocity field, and  $V^N$  is the magnitude of the interface velocity along the normal axis. In practical usage, some of the terms are neglected. In this study, perfusion (rotation of the liquid around the bulk sample) is not considered, and the degradation rate does not depend on the curvature of the interface. As a result, by assuming that the interface moves in normal direction only, Eq. 7.13 can be simplified to

$$\frac{\partial \phi}{\partial t} + V^N |\nabla \phi| = 0 \quad (7.14)$$

where  $V^N$  is depicted in Fig. 7.2. The RankineHugoniot equation can be used to calculate the interface velocity in mass transfer problems [60]:

$$\{J(x, t) - (c_{\text{sol}} - c_{\text{sat}}) V(x, t)\} \cdot n = 0 \quad (7.15)$$

in which  $J$  is the mass flux,  $c_{\text{sol}}$  is the concentration of the material in the bulk part (i.e. its density), and  $c_{\text{sat}}$  is the concentration at which the material (here, the ions) saturates through the medium. So, for the investigated Mg

degradation problem, Eq. 7.15 will be:

$$D_{\text{Mg}}^e \nabla_n C_{\text{Mg}} - ([\text{Mg}]_{\text{sol}} - [\text{Mg}]_{\text{sat}}) V^N = 0. \quad (7.16)$$

Inserting the obtained velocity of Eq. 7.16 into Eq. 7.14 and considering the direction of the shrinkage velocity, which is in the opposite direction of the surface normal vector, yields

$$\frac{\partial \phi}{\partial t} - \frac{D_{\text{Mg}}^e \nabla_n C_{\text{Mg}}}{[\text{Mg}]_{\text{sol}} - [\text{Mg}]_{\text{sat}}} |\nabla \phi| = 0. \quad (7.17)$$

Eq. 7.17 is the final formulation of the level set equation in the current study, which alongside Eqs. 7.10 and 7.11 forms the mathematical model of degradation of Mg with a moving interface. Eq. 7.17 contributes indirectly to the evolution of Eqs. 7.10 and 7.11 as it defines the boundary, the zero iso-contour of the  $\phi$  function, on which the boundary conditions of the equations are applied.

### 7.3 Methodology of model implementation

The developed mathematical model comprised of Eqs. 7.10, 7.11, and 7.17 cannot be solved using analytical techniques. The alternative approach in these scenarios is solving the derived PDEs numerically. In this study, we used a combination of finite element and finite difference methods to solve the aforementioned equations. In the developed numerical model, the PDEs are solved one by one, each of which is a linear equation, so the model implementation follows the principles of solving linear systems. In the following section, only the process to obtain the solution of Eq. 7.10 is described in detail, but the other PDEs were solved using the same principle. Although the adopted finite element method is standard, we elaborate on its derivation to clarify the bottlenecks of the later-discussed implementation.

### 7.3.1 Finite element discretization (bottleneck of the algorithm)

In order to solve Eq. 7.10 numerically, we used a finite difference scheme for the temporal term and a finite element formulation for the spatial terms. For simplicity of writing, notations of variables are changed, so  $C_{\text{Mg}}$  is represented as  $u$  (the main unknown state variable to find),  $C_{\text{Film}}$  is denoted by  $p$ ,  $[\text{Cl}]$  is denoted by  $q$ , and the saturation term  $(1 - \frac{F}{F_{\max}})$  is denoted by  $s$ . By doing this, Eq. 7.10 can be written as

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - k_1 s u + k_2 p q^2. \quad (7.18)$$

To obtain the finite element formulation, the weak form of derived PDE is required. In order to get this, we define a space of test functions and then, multiply each term of the PDE by any arbitrary function as a member of this space. The test function space is

$$\mathcal{V} = \left\{ v(\mathbf{x}) | \mathbf{x} \in \Omega, v(\mathbf{x}) \in \mathcal{H}^1(\Omega), \text{ and } v(\mathbf{x}) = 0 \text{ on } \Gamma \right\} \quad (7.19)$$

in which the  $\Omega$  is the domain of interest,  $\Gamma$  is the boundary of  $\Omega$ , and  $\mathcal{H}^1$  denotes the Sobolev space of the domain  $\Omega$ , which is a space of functions whose derivatives are square-integrable functions in  $\Omega$ . The solution of the PDE belongs to a trial function space, which is similarly defined as

$$\mathcal{S}_t = \left\{ u(\mathbf{x}, t) | \mathbf{x} \in \Omega, t > 0, u(\mathbf{x}, t) \in \mathcal{H}^1(\Omega), \text{ and } \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma \right\}. \quad (7.20)$$

Then, we multiply Eq. 7.18 to an arbitrary function  $v \in \mathcal{V}$ :

$$\frac{\partial u}{\partial t} v = \nabla \cdot (D \nabla u) v - k_1 s u v + k_2 p q^2 v. \quad (7.21)$$

Integrating over the whole domain yields:

$$\int_{\Omega} \frac{\partial u}{\partial t} v d\omega = \int_{\Omega} \nabla \cdot (D \nabla u) v d\omega - \int_{\Omega} k_1 s u v d\omega + \int_{\Omega} k_2 p q^2 v d\omega. \quad (7.22)$$

The diffusion term can be split using the integration by parts technique:

$$\int_{\Omega} \nabla \cdot (D \nabla u) v d\omega = \int_{\Omega} \nabla \cdot [v(D \nabla u)] d\omega - \int_{\Omega} (\nabla v) \cdot (D \nabla u) d\omega \quad (7.23)$$

in which the second term can be converted to a surface integral on the domain boundary by applying the Green's divergence theory:

$$\int_{\Omega} \nabla \cdot [v(D \nabla u)] d\omega = \int_{\Gamma} Dv \frac{\partial u}{\partial n} d\gamma. \quad (7.24)$$

For the temporal term, we use the finite difference method and apply a first-order backward Euler scheme for discretization, which makes it possible to solve the PDE implicitly:

$$\frac{\partial u}{\partial t} = \frac{u - u^n}{\Delta t} \quad (7.25)$$

where  $u^n$  denotes the value of the state variable in the previous time step (or initial condition for the first time step). Inserting Eqs. 7.23, 7.24, and 7.25 into Eq. 7.22 yields:

$$\int_{\Omega} \frac{u - u^n}{\Delta t} v d\omega = \int_{\Gamma} Dv \frac{\partial u}{\partial n} d\gamma - \int_{\Omega} D \nabla u \cdot \nabla v d\omega - \int_{\Omega} k_1 s u v d\omega + \int_{\Omega} k_2 p q^2 v d\omega. \quad (7.26)$$

The surface integral is zero because there is a no-flux boundary condition on the boundary of the computational domain (defined in the trial function space according to Eq. 7.20). By reordering the equation, we get the weak form of Eq. 7.18:

$$\int_{\Omega} u v d\omega + \int_{\Omega} \Delta t D \nabla u \cdot \nabla v d\omega + \int_{\Omega} \Delta t k_1 s u v d\omega = \int_{\Omega} u^n v d\omega + \int_{\Omega} \Delta t k_2 p q^2 v d\omega. \quad (7.27)$$

So, the problem is finding a function  $u(t) \in \mathcal{S}_t$  such that for all  $v \in \mathcal{V}$  Eq. 7.27 would be satisfied. By defining a linear functional  $(f, v) = \int_{\Omega} f v d\omega$  and encapsulating the independent concentration terms into  $f^n = pq^2$ , Eq. 7.27 can be simplified as:

$$(u, v)[1 + \Delta t k_1 s] + \Delta t (D \nabla u, \nabla v) = (u^n, v) + \Delta t (f^n, v) \quad (7.28)$$

which can be further converted to the common form of the weak formulation of time-dependent reaction-diffusion PDEs by multiplying to a new coefficient

$$\alpha = \frac{1}{1 + \Delta t k_1 s}:$$

$$(u, v) + \alpha \Delta t (D \nabla u, \nabla v) = \alpha (u^n, v) + \alpha \Delta t (f^n, v). \quad (7.29)$$

One can approximate the unknown function  $u$  in Eq. 7.29 by  $u(x) \approx \sum_{i=0}^N c_i \psi_i(x)$ , where the  $\psi_i$  are the basis functions used to discretize the function space, and  $c_0, \dots, c_N$  are the unknown coefficients. The finite element method uses Lagrange polynomials as the basis function and discretizes the computational domain using a new function space  $\mathcal{V}_h$  spanned by the basis functions  $\{\psi_i\}_{i \in \mathcal{I}_s}$ , in which  $\mathcal{I}_s$  is defined as  $\mathcal{I}_s = \{0, \dots, N\}$ , where  $N$  denotes the degrees of freedom in the computational mesh. The computational mesh discretizes the space into a finite number of elements, in each of which the  $\psi_i$  is non-zero inside the  $i$ th element and zero everywhere else. In this study, 1st order Lagrange polynomials were used as the basis functions to define the finite element space.

For 1D elements, a 1st order Lagrange polynomial for the  $i$ th element with the width of  $h$  can be written as:

$$\psi_i(x) = \begin{cases} 0 & x < x_{i-1} \\ (x - x_{i-1}) / h & x_{i-1} \leq x < x_i \\ 1 - (x - x_i) / h & x_i \leq x < x_{i+1} \\ 0 & x \geq x_{i+1} \end{cases}. \quad (7.30)$$

A similar approach can be applied to define the basis function space in 2D and 3D spaces.

In order to derive a linear system of equations for obtaining the unknown coefficients  $c_j$ , we define

$$u = \sum_{j=0}^N c_j \psi_j(x), \quad u^n = \sum_{j=0}^N c_j^n \psi_j(x) \quad (7.31)$$

as the definition of the unknown function  $u$  and its value in the previous time step  $u^n$ . We then insert it into Eq. 7.29, which yields the following equation for each degree of freedom  $i = 0, \dots, N$ , where the test functions are selected as

$$v = \psi_i:$$

$$\sum_{j=0}^N (\psi_i, \psi_j) c_j + \alpha \Delta t \sum_{j=0}^N (\nabla \psi_i, D \nabla \psi_j) c_j = \sum_{j=0}^N \alpha (\psi_i, \psi_j) c_j^n + \alpha \Delta t (f^n, \psi_i). \quad (7.32)$$

Eq. 7.32 is a linear system

$$\sum_j A_{i,j} c_j = b_i \quad (7.33)$$

with

$$A_{i,j} = (\psi_i, \psi_j) + \alpha \Delta t (\nabla \psi_i, D \nabla \psi_j) \quad (7.34)$$

$$b_i = \sum_{j=0}^N \alpha (\psi_i, \psi_j) c_j^n + \alpha \Delta t (f^n, \psi_i) \quad (7.35)$$

which can also be rewritten as

$$(M + \alpha \Delta t K) c = \alpha M c_1 + \alpha \Delta t f. \quad (7.36)$$

$M$  (which traditionally is called the mass matrix),  $K$  (which traditionally is called the stiffness matrix),  $f$ ,  $c$ , and  $c_1$  are defined as

$$\begin{aligned} M &= \{M_{i,j}\}, \quad M_{i,j} = (\psi_i, \psi_j), \quad i, j \in \mathcal{I}_s \\ K &= \{K_{i,j}\}, \quad K_{i,j} = (\nabla \psi_i, D \nabla \psi_j), \quad i, j \in \mathcal{I}_s \\ f &= \{f_i\}, \quad f_i = (f(x, t_n), \psi_i), \quad i \in \mathcal{I}_s \\ c &= \{c_i\}, \quad i \in \mathcal{I}_s \\ c_1 &= \{c_i^n\}, \quad i \in \mathcal{I}_s \end{aligned} \quad (7.37)$$

By solving Eq. 7.33 and substituting the obtained  $c$  in Eq. 7.31,  $u$  ( $C_{\text{Mg}}$  in the example in this study) can be calculated in the current time step. As stated before, the same approach can be applied to Eq. 7.11 and Eq. 7.17 to get  $C_{\text{Film}}$  and  $\phi$ . This procedure is repeated in each time step to compute the values of  $C_{\text{Mg}}$ ,  $C_{\text{Film}}$ , and  $\phi$  over time.

A common practice to save time for solving Eq. 7.33 for a constant time step size is to compute the left-hand side matrix ( $A$  in Eq. 7.34) once and compute only

the right-hand side vector of the equation at each time iteration. But in this case, although the time step size is fixed, due to the presence of the  $\alpha$  coefficient, the matrix changes along the time. The  $\alpha$  coefficient is not constant and should be updated in each time step because it depends on the penalization term  $s$  (which is a function of the concentration of the film as can be seen by comparing Eq. 7.10 and Eq. 7.18). In addition to this, the diffusion coefficient is not constant (Eq. 7.12), making the second term in Eq. 7.34 non-constant even in the absence of  $\alpha$  coefficient. Consequently, the left-hand side matrix of the Eq. 7.33 cannot be computed before the start of the main time loop, and computing it in each time step is an extra but inevitable computational task in comparison to similar efficient and high-performance finite element implementations. This contributes to a slower algorithm for solving the aforementioned PDEs.

### 7.3.2 Implementation and parallelization

The model was implemented in FreeFEM [61], which is an open-source PDE solver to facilitate converting the weak formulation (Eq. 7.27) to a linear system  $Ax = b$  (with  $A$  from Eq. 7.34 and  $b$  from Eq. 7.35). The computational mesh was generated using Netgen [66] in the SALOME platform [67] by a set of linear tetrahedral elements, and all the other preprocessing steps were performed in FreeFEM. The mesh was adaptively refined on the material-medium interface in order to increase the accuracy of the level set model. Postprocessing of the results was carried out using Paraview [83].

Computing the diffusion solely in the medium domain causes oscillations close to the interface, and to prevent this, the mass lumping feature of FreeFEM was employed. In this technique, the desired mass matrix is handled node-wise and not element-wise. Technically speaking, this means that the state variable is stored in the mesh nodes, and although this is the natural formulation in the finite difference method, it requires artificial modification in the standard finite element formulation [84]. The mass lumping feature of FreeFEM applies a quadratic formula at the vertices of elements to make the mass matrix diagonal, which contributes positively to the convergence of the solution.

The main parallelization approach for the current study was domain decomposition, in which the mesh is split into smaller domains (can be overlapping or non-overlapping), and the global solution of the linear system is achieved by solving the problem on each smaller local mesh. What really matters in this approach is providing virtual boundary conditions to the smaller sub-

domains by ghost elements, transferring neighboring sub-domain solutions [85]. As a result, a high-performance parallelism is feasible by assigning each sub-domain to one processing unit.

In computational science, preconditioning is widely used to enhance the convergence, which means instead of directly working with a linear system  $Ax = b$ , one can consider the preconditioned system [86]:

$$M^{-1}Ax = M^{-1}b \quad (7.38)$$

in which the  $M^{-1}$  is the preconditioner. In the current study, we considered this approach for both the domain composition and the solution of the linear system. We opted to use an overlapping Schwarz method for domain decomposition, in which the mesh is first divided into a graph of  $N$  non-overlapping meshes using METIS (or ParMETIS) [87]. Then, by defining a positive number  $\delta$ , the overlapping decomposition  $\{\mathcal{T}_i^\delta\}_{1 \leq i \leq N}$  can be created recursively for each sub-mesh  $\{\mathcal{T}_i\}_{1 \leq i \leq N}$  by adding all adjacent elements of  $\mathcal{T}_i^{\delta-1}$  to it. Then, the finite element space  $\mathcal{V}_h$  (Eq. 7.19) can be mapped to the local space  $\{\mathcal{V}_i^\delta\}_{1 \leq i \leq N}$  by considering the restrictions  $\{R_i\}_{1 \leq i \leq N}$  and a local partition of unity  $\{D_i\}_{1 \leq i \leq N}$  such that:

$$\sum_{j=1}^N R_j^\top D_j R_j = I_{n \times n} \quad (7.39)$$

where  $I$  and  $n$  denote identity matrix and the global number of unknowns, respectively [88].

In this study, we decomposed the mesh by using the one-level preconditioner Restricted Additive Schwarz (RAS):

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^N R_i^\top D_i A_i^{-1} R_i \quad (7.40)$$

in which  $\{A_i\}_{1 \leq i \leq N}$  is the local operator of the sub-matrices [88]. For this purpose, we took advantage of the HPDDM (high-performance domain decomposition methods) package interface in FreeFEM [65]. The partitioned mesh is shown in Fig. 7.3. The effect of the construction of these local sub-domains on the sparsity pattern of the global matrix is also depicted in Fig. 7.4. The global matrix is a sparse matrix according to Eq. 7.34 and the definition of the basis function  $\psi$ .

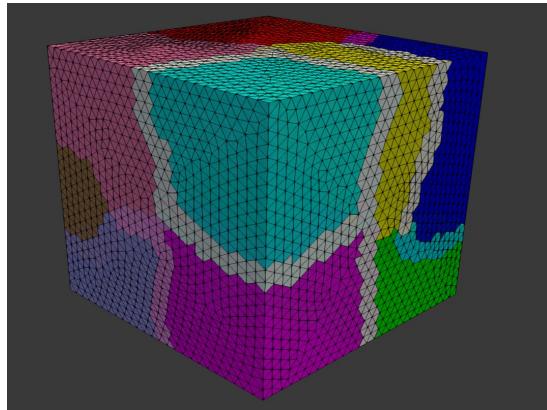


Figure 7.3: Overlapping domain decomposition in the current study. Each color shows a separate sub-domain, and the narrow lighter bands are the overlapped regions.

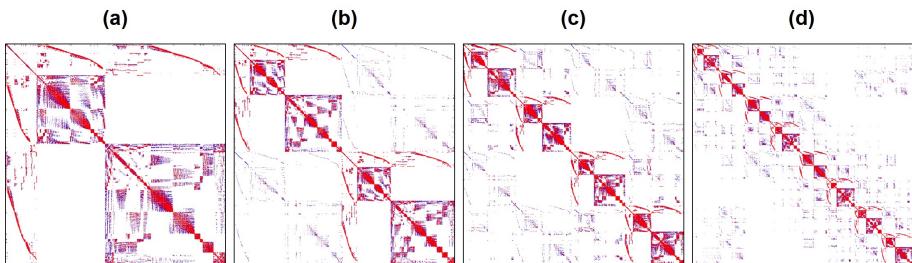


Figure 7.4: Comparison of the sparsity patterns (highlighting non-zero elements) of the global matrix  $A$  for a different number of decomposed domains a: 1 domain b: 2 sub-domains c: 4 sub-domains d: 8 sub-domains.

Generally, two categories of methods have been used to solve a large linear system of equations on parallel machines: direct solvers (e.g. Multifrontal Massively Parallel Sparse, MUMPS [89]) and iterative solvers (e.g. Generalized Minimal Residual Method, GMRES [63]). While direct solvers are quite robust, they suffer from the memory requirement problem on large systems. Inversely, iterative solvers are quite efficient on memory consumption, but similar to other iterative approaches, they are not very reliable in some

cases [90]. Direct solvers modify the matrix by factorization (e.g. Cholesky decomposition), but an iterative solver does not manipulate the matrix and works solely using basic algebraic operations. However, for an efficient usage of iterative solvers, a proper preconditioner is crucial [90]. By evaluating and comparing the performance of the aforementioned methods for the current model, we decided to use an iterative approach using the Krylov subspaces (KSP) method, in which we preconditioned the equation using a proper preconditioner (Eq. 7.38) and then solved it with an iterative solver.

Krylov methods have been frequently used by researchers as robust iterative approaches to parallelism [91]. What matters in this regard is ensuring proper scaling of the parallelized algorithm for both the assembling of the matrices and the solution of the linear system of equations. One good solution to this challenge is taking advantage of HPC-ready mathematical libraries to achieve efficient distributed-memory parallelism through the Message Passing Interface (MPI). In the current study, we used the PETSc (Portable, Extensible Toolkit for Scientific Computation) library [64], which provides a collection of high-performance preconditioners and solvers for this purpose.

In order to yield the highest performance, a variety of different combinations of KSP types and preconditioners were evaluated, such as Conjugate Gradients (CG) [92], Successive Over-Relaxation (SOR) [93], block Jacobi, and Algebraic Multigrid (AMG) [94], to name a few. The performance tests results are presented in the supplementary materials of this paper. The best performance for the reaction-diffusion system model was achieved using the HYPRE preconditioner [62] and the GMRES solver [63]. This was the combination used for all the performance analysis tests.

### 7.3.3 Level set issues

As mentioned before, in order to track the interface of the bulk material and the surrounding fluid, an implicit signed distance function is defined as the solution of Eq. 7.17. This equation can be solved using the aforementioned finite element discretization, but in a practical implementation, there are usually a couple of problems associated with this PDE.

The first issue is defining  $D_{\text{Mg}}^e$  and  $\nabla_n C_{\text{Mg}}$  on the moving interface. To ensure correct boundary conditions for Eq. 7.16, the value of  $C_{\text{Mg}}$  is set constant on the whole bulk material by using the penalty method. As a result, the implicit

interface is not necessarily aligned on the computational mesh. Although this is a beneficial fact for the interface tracking, it inserts the problem of overestimation of  $C_{\text{Mg}}$  on the nodes close to the interface, which makes it difficult to calculate  $\nabla_n C_{\text{Mg}}$  on these nodes correctly. The same problem exists for calculating  $D_{\text{Mg}}^e$ . To overcome this issue, the values of  $C_{\text{Mg}}$  and  $D_{\text{Mg}}^e$  are calculated at the distance  $h$  from the interface in the normal direction (towards the medium), where  $h$  is the edge size of the smallest element of the computational mesh.

The next issue is a well-known problem of the level set method: if the velocity of the interface is not constant (as in Eq. 7.13), the level set function  $\phi$  may become distorted by having too flat or too steep gradients close to the moving front. This could cause unwanted movements of the interface. The problem becomes even worse when the distance function is advected. A solution to this issue is re-initializing the distance function in each time step (re-distancing), but this operation requires solving a new PDE. From numerical investigations, it has been observed that this operation inserts new errors in the numerical computation of the level set equation [95]. This can be resolved by improving the method of reconstruction of the distance function [95].

However, re-initialization results in another issue on a massively parallel implementation: as the mesh is partitioned into smaller sub-meshes, it is not feasible anymore to evaluate the distance to the interface globally on each sub-domain. As a result, the inverse process of domain decomposition should be taken to assemble the mesh again. This can be done by the restriction matrix and the partition of unity (defined in Eqs. 7.39 and 7.40), but it is rather a very inefficient procedure regarding the parallelization of the simulation and results in a long execution time in each time step.

In the current study, the distance function  $\phi$  was initialized only once at the beginning of the simulation. The re-initialization process was unnecessary in this case because according to Eq. 7.17, the distance function is advected only in the regions where there is a gradient of the concentration of Mg ions, which means that advection is applied only on the regions close to the interface in the medium. This prevented the whole distance function of being distorted, and as a result, it was not required to re-initialize it in each time step. This also removed the need for inverting the decomposition process.

### 7.3.4 Simulation setup

In order to verify the performance of the developed model, a degradation experiment was reconstructed in-silico, in which the degradation of a block of Mg (with the size of  $13\text{mm} \times 13\text{mm} \times 4\text{mm}$ ) was investigated in a simulated body fluid solution. All the experimental parameter data (used to setup the simulation), as well as the degradation rates (used to calibrate and validate the numerical model) were extracted from [16].

As can be seen in Eqs. 7.1 and 7.2, each mole removed from the Mg block corresponds to one mole of the produced hydrogen. As a result, instead of a direct measurement of mass loss, one can collect and measure the amount of produced hydrogen to monitor the degradation rate. This is a common way of reporting degradation in this type of studies [96]. In order to get this quantity out of the developed model, we used the level set model output. The total mass loss of Mg at each desired time can be calculated based on the movement of the corrosion front:

$$\text{Mg}_{\text{lost}} = \int_{\Omega_+(t)} \text{Mg}_{\text{solid}} dV - \int_{\Omega_+(0)} \text{Mg}_{\text{solid}} dV \quad (7.41)$$

where  $\Omega_+(t) = \{\mathbf{x} : \phi(\mathbf{x}, t) \geq 0\}$ . It is worth noting that this integration should be performed by ignoring the ghost elements generated in the mesh partitioning process, otherwise the calculated material loss will be higher than the real value. Then, the amount of formed hydrogen gas can be calculated based on the ideal gas law:

$$H_f = \frac{\text{Mg}_{\text{lost}}}{\text{Mg}_{\text{mol}}} \frac{RT}{PA} \quad (7.42)$$

in which  $R$  is the universal gas constant,  $P$  is the pressure,  $T$  is the solution temperature,  $A$  is the exposed corrosion surface area (which can be computed using the level set function), and  $\text{Mg}_{\text{mol}}$  is the molar mass of Mg. Plotting a comparison of the predicted and experimentally obtained values of hydrogen can show the overall validity of the mathematical model because both the diffusion-reaction equations and the level set equation contribute to the prediction made by the computational model.

The geometry of the simulation experiment is depicted in Fig. 7.5. Based on this geometry, an Eulerian computational mesh was constructed by generating

tetrahedral elements on the whole domain, including the Mg block and the medium. This resulted in 830,808 elements with a total of 143,719 DOFs for each PDE (Eqs. 7.10, 7.11, and 7.17), which indicates the size of matrix  $A$  in Eq. 7.34. Model parameters and material properties were obtained from [49]. The diffusion coefficient of Mg was calculated using an inverse problem setup in which a Bayesian optimization process [68] was used to run the simulation code multiple times and minimize the difference of the model output and the experimental data reported by [16]. A time step convergence study was performed to measure the sensitivity of the model to the time stepping parameter, and based on the results, the time step value was set to 0.025 hours.

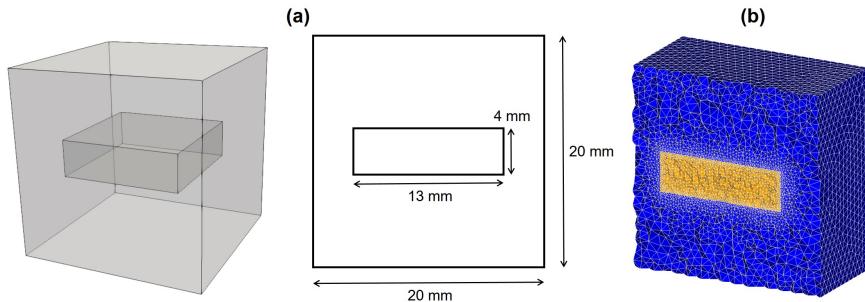


Figure 7.5: Representation of the experimental set-up simulated to perform numerical validation of the developed model and evaluate parallel performance. a) A cuboid of Mg (with the size of  $13\text{mm} \times 13\text{mm} \times 4\text{mm}$ ) is floating inside a simulated body fluid solution to investigate the degradation process, b) a cross-section of the computational mesh, refined on the metal-medium interface to increase the interface capturing accuracy.

### 7.3.5 Performance analysis

To investigate the performance and scaling behavior of the implemented parallel code, we conducted a set of weak-scaling and strong-scaling tests on the computational model. To do this, the time required to solve each PDE in each time step was measured in a simulation. This acted as a rough estimation of the time required in each time step because it ignores all the other factors contributing to speedup results such as communication costs, load imbalance, limited memory bandwidth, and parallelization-caused overhead.

Weak-scaling was evaluated by dividing the computational domain into smaller sub-domains (each of which was  $\frac{1}{16}$  of the whole domain, Fig. 7.6) and conducting simulation experiments with 1, 2, 4, and 8 computational cores in a way that the number of processors corresponded to the number of employed sub-domains. In Fig. 7.6 the upper row shows different domains as an accumulation of the smaller divisions, and the lower row shows the corresponding domain decomposition for parallel computing by depicting each processing unit in a different color. In fact, it demonstrates the concept of increasing the number of MPI processing units as we increase the size of the problem.

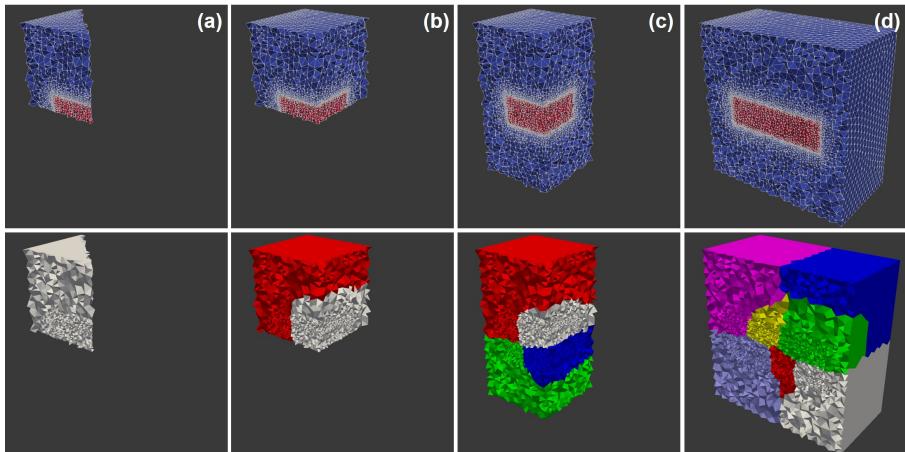


Figure 7.6: Models used for weak-scaling, in which the number of elements was doubled each time while doubling the number of computational cores. Upper row: actual computational domain in which colors show the medium (blue) and the material block (red). Lower row: domain decomposition for parallelization, colors show different decomposed mesh parts (distributed to different MPI processing units). Each column corresponds to a different simulation with a: 1 MPI unit, b: 2 MPI units, c: 4 MPI units, and d: 8 MPI units.

After calculating the speedup of each test (by comparing the differences in execution time), we can use Gustafsons law [97] to calculate the sequential and parallelizable portion of computation in the current implementation in weak-scaling evaluation:

$$\text{Speedup} = f + (1 - f) \times N \quad (7.43)$$

where  $N$  is the total number of computational cores,  $f$  is the fraction of operations in the computation that are sequential, and as a result,  $1 - f$  is the fraction of the execution time spent on the parallelizable part.

The strong-scaling evaluation was performed using the entire domain. The evaluation was done using 1, 8, 16, 40, 60, 90, 200, and 300 MPI cores. In strong-scaling, Amdahls law [98] is used to calculate the portion of the algorithm that runs in parallel:

$$\text{Speedup} = \frac{1}{f + \frac{1-f}{N}} \quad (7.44)$$

in which the parameters are the same as Eq. 7.43.

### 7.3.6 Compute environment

Simulations were conducted on the VSC (Flemish Supercomputer Center) supercomputer with the availability of Intel CPUs in three different micro-architectures: Ivy Bridge, Haswell, and Skylake. Due to a better performance, the strong and weak-scaling measurements were solely performed on the Skylake nodes. On this supercomputer, we made use of 3 nodes, 36 cores each, with 576 GB of the total memory, each node holding 2 Intel Xeon Gold 6132 CPUs with a base clock speed of 2.6 GHz. The supercomputer uses CentOS 7.6.1810 with kernel version 3.10.0. For interprocess communication, Intel's MPI implementation 2018 was used.

## 7.4 Results

### 7.4.1 Numerical simulation results

The performed numerical simulation produces the output of three main quantities: the concentration of the Mg ions in the medium (as the solution of Eq. 7.10), the concentration of the protective film (as the solution of Eq. 7.11), and the level set function values at each element (as the solution of Eq. 7.17). In addition to this, a quantitative prediction of the mass loss is also generated according to Eqs. 7.41 and 7.42.

In order to have quantitative predictions, the coefficients of Eqs. 7.10 and 7.11 (diffusion rates and reaction rates) should be calibrated using an inverse problem. Fig. 7.7 shows the results produced by the computational model after this parameter estimation stage. A narrow layer of the protective film is formed on the surface of the Mg block, and the volume of produced hydrogen gas is compared with values obtained from experiments. Additionally, by plotting the zero iso-contour of the level set function, we can obtain the shape of the material block as it degrades during the degradation process (i.e. tracking the moving corrosion front). This is depicted by the grey surface in Fig. 7.7.

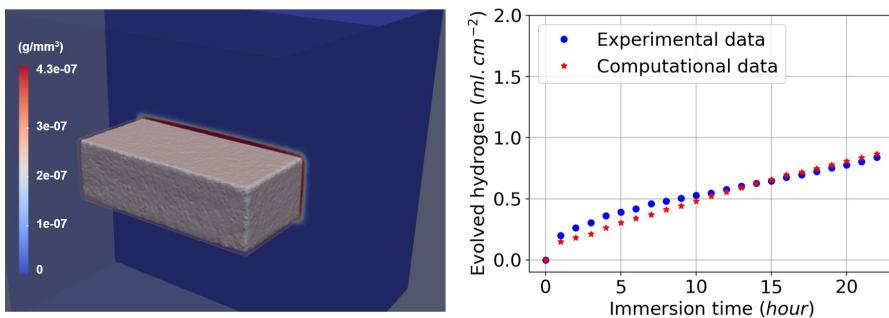


Figure 7.7: Numerical simulation result. Left: formation of a protective layer on the surface of the Mg block (red region). Right: comparison of the produced hydrogen (a surrogate for the material loss) in the computational model and the experimental data, which is a validation of the full model as both the reaction-diffusion equations and the level-set equation are involved in the computation of this quantity.

#### 7.4.2 Weak and strong scaling results

Weak-scaling results are plotted in Fig. 7.8, in which the execution time of each time step is broken down into the time spent on each PDE. The results show good scalability of the parallel implementation.

Speedup and parallel efficiency of the weak-scaling experiment is plotted in Fig. 7.9. By fitting a curve based on the Gustafson equation (Eq. 7.43) on the obtained results (Fig. 7.9), the sequential proportion of the current implementation was calculated to be 18%, which means that 82 percent of the code can be parallelized, which is a proper but not an ideal scalability.

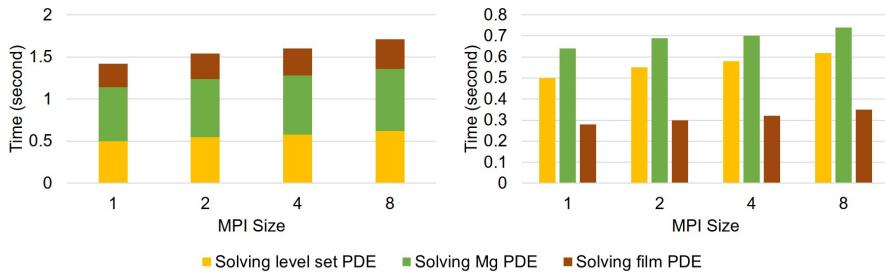


Figure 7.8: Weak-scaling test result. Results are broken down into contributions for each PDE, which are plotted cumulatively and separately in the left and right plot, respectively.

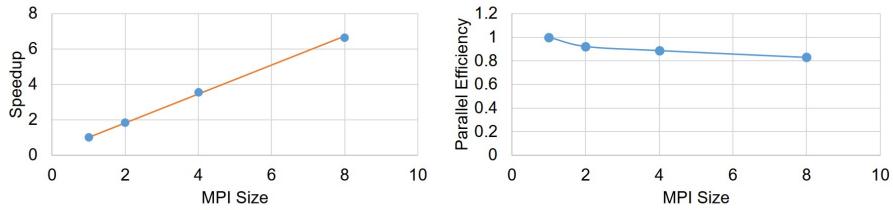


Figure 7.9: Speed-up and parallel efficiency of the weak-scaling experiment. The orange line in the left plot shows the fitted curve based on the Gustafson equation.

The strong-scaling results are plotted in Fig. 7.10, which shows a better scalability in comparison to the weak-scaling test. For a better representation, exact measured values are presented in Table 7.1.

Table 7.1: Strong-scaling test result, presented by the execution time of each PDE in simulations with a different number of employed MPI cores.

MPI Size	1	8	16	40	60	90	200	300	
Solution time of each time step (s)	LS PDE	9	1.39	0.75	0.36	0.26	0.19	0.11	0.07
	Mg PDE	13.04	1.76	0.94	0.46	0.31	0.22	0.12	0.09
	Film PDE	6.38	0.84	0.45	0.21	0.14	0.09	0.05	0.04
Total time (s)		28.42	3.99	2.14	1.03	0.71	0.5	0.28	0.2

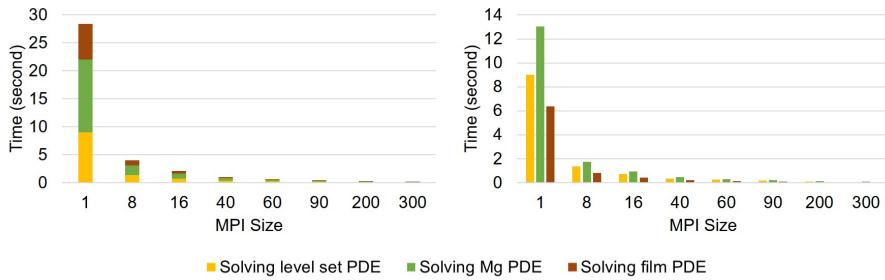


Figure 7.10: Strong-scaling test result. Results are broken down into contributions for each PDE, which are plotted cumulatively and separately in the left and right plot, respectively.

Similar to weak-scaling results, Fig. 7.11 demonstrates the speedup and parallel efficiency of the developed code for strong-scaling evaluation. From the results, it is obvious that increasing the number of cores leads to a better performance but a lower efficiency. By fitting Amdahls equation (Eq. 7.44) on the obtained speedup results (Fig. 7.11),  $f$  was obtained as 0.01, which means in strong-scaling terms that 99% of the code can run in parallel.

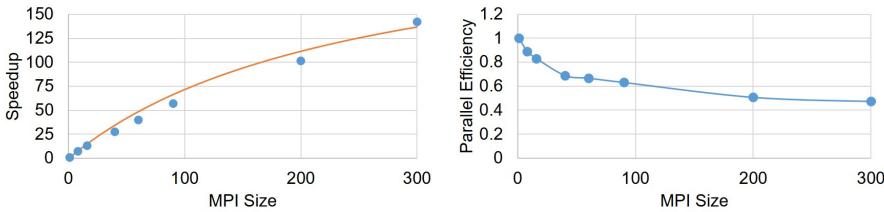


Figure 7.11: Speed-up and parallel efficiency of the strong-scaling experiment. The orange line in the left plot is the fitted equation based on the Amdahl rule.

## 7.5 Discussion

In this investigation, the derivation and implementation of a reaction-diffusion model with moving boundaries were presented. Such an approach finds application in many scientific and engineering problems. The target application in the current work was the degradation of a bulk metal cuboid in a liquid

environment, specifically Mg in an aqueous ion solution as a representative for temporary medical devices. The simulations were based on the corrosion of Mg metal to Mg ions to form a film of Mg hydroxide that partially protects the metal block from further degradation except where this film is impacted by reaction with other ions in the environment (such as chloride ion). The reactive moving boundary problem was cast in the form of equations in which the change of the concentrations of the different chemical components is represented by parabolic PDEs. The coupled equations depend on several kinetic constants that have been calibrated from experiments. The moving interface between the metal bulk and the liquid phase was described by an implicit function using the level set method. The derivation led to equations that require the use of numerical techniques for which a combination of finite difference and finite element methods was implemented. As the required high accuracy on the moving interface results in an increase in computation time, parallelization was crucial for the computational model to decrease the execution time of the simulations. The results of the total execution time in each time step (Table 7.1) clearly indicate that without the parallelization, the simulation of the model is slow and as a result, less interactable for real-world simulation analyses. Considering the properly employed parallelization, the computational time has been decreased noticeably for the investigated case-study.

The output of the conducted numerical simulation demonstrates that the developed mathematical model is capable of capturing the degradation interface movement and of modeling of the underlying chemical phenomena. The predicted mass loss is in line with the experimental results, and the simulated corrosion behavior is as expected for such a system. It is worth noting that the chosen system is highly idealized as a model for medical devices. A more realistic chemical environment would contain many more species that play a role in the formation of either soluble ions or the protective film. Moreover, in real-world scenarios, corrosion occurs in a more complex way than the simplified one described in this paper, which will have a significant influence on the local concentration of ions in the regions close to the solid surface. Nevertheless, the developed framework is capable of capturing these physical and chemical phenomena in the future by simply adding the appropriate terms to the base PDEs without any major changes in the computational model. Furthermore, although it requires some changes to the parallelization approach, the addition of the fluid flow around the block is feasible by adding convective terms to form a reaction-diffusion-convection system. Such a system

can be used to model relevant systems such as experimental bioreactor setups in biology and medical sciences.

The parallel algorithm was implemented using a domain decomposition method. Standard domain decomposition preconditioners, such as restricted additive Schwarz, are widely used for parallel implementation of computational models. In a parallel implementation, such preconditioners bring the benefit of relatively low communication costs [86]. Beside this, the formed linear system of equations in each partition of the mesh was solved using Krylov methods by taking advantage of the highly-efficient preconditioners and iterative solvers of the PETSc library. According to the obtained results, the employed parallelization approach of the current study yields reasonable scaling with respect to the available computational resources (or the number of sub-domains). Out of multiple evaluations, the best performance was achieved using the preconditioner/solver combination of HYPRE/GMRES, which is in agreement with findings in more specific studies in this regard [99].

To evaluate the scaling performance of the implemented parallelism, a set of weak and strong scaling tests was conducted. In weak-scaling, the main approach is changing the problem size proportional to the change in the available computing resources. In an ideal parallelization, we expect that the speedup remains the same for all the setups because we provide double resources as we double the size of the problem. In strong-scaling, the size of the problem remains constant, but the number of computing units increases. So, in an ideal case, we should observe a double speedup as the number of computing units doubles. By fitting Gustafson's and Amdahl's laws on the scaling test results (Figs. 7.9 and 7.11), the maximum parallelizable portion of the code was calculated to be 82% and 99% for the weak-scaling and strong-scaling tests, respectively. This is a reasonable theoretical scaling for both cases.

The obtained scaling behavior is similar to other conducted studies for diffusion or diffusion-convection systems [100, 101], in which the efficiency of the parallelization decreases with increasing the number of available computational resources. The reason behind this behavior in the current model lies in the mesh partitioning process. Indeed, the mesh is partitioned into semi-equal partitions, each of which has the same number of elements, but the main computation is only carried out on the nodes located outside the degrading material block (i.e. in the medium). In other words, the computational resources assigned to the nodes inside the material bulk do not contribute significantly to the simulation. This limitation can be prevented

by modifying the mesh generation process in a way that a lower number of elements be generated inside the material block, but doing this requires remeshing of the interior region as the moving interface approaches it, which imposes even more complexity to the algorithm due to the partitioned mesh. Another bottleneck of the current model, as discussed before, routed in the non-constant right-hand matrix of the linear system (Eq. 7.33), which requires computing the  $A$  matrix (Eq. 7.34) in each time step and leads to a slower execution time.

One important point in this regard is that the way that the results are interpreted does not necessarily imply the true scaling behavior of the system. Indeed, it is more like a surrogate model of the system performance. The correct methodology for obtaining true scaling factors is rather starting from an analysis of the code and time used in each routine for a non-parallel run. Then, based on the fraction of routines that are possible to execute in parallel, one can get a theoretical limit for the speedup. This will be reduced by practical limitations such as load balancing and communication costs of the network. Since it is a theoretical limit, it is not fully correct to ignore those extra parts and use the execution time to invert the relation to predict the fraction of the code that is parallel. However, for a complex computational model like the one that was developed in the current study, doing such a measurement of each routine is very difficult due to the complexity of the orchestrated libraries and tools. As a result, we were limited to use the roughly approximated speedup limit to evaluate the scaling of the constructed model. Regarding the scalability results, it is worth mentioning that although having studies with thousands of MPI ranks is more common in this field, due to the limitation we faced in accessing computational resources, the maximum number of employed cores were limited to 300. The goal of the current study was to demonstrate the scalability of the developed model on massively parallel systems, and the behavior of the model in moving from 90 cores to 300 shows the consistency in the performed performance analysis. As a result, we expect to see the same scalability behavior for problems in a larger scale with a higher number of employed computing nodes.

## 7.6 Conclusion

In this work, a mathematical model of a reaction-diffusion system with a moving front was constructed, and the corresponding computational model

was implemented using the finite element method. In order to correlate the diffusion phenomenon to the moving boundary position, high numerical accuracy is necessary at the diffusion interface, which requires a finer discretization of space near the moving front. This leads to an expensive computational model, which makes employing HPC techniques crucial in order to improve the simulation execution time. To this end, a high-performance domain decomposition approach was employed to partition the mesh and distribute the workload to available computing resources. Additionally, an efficient preconditioner/solver combination for reaction-diffusion PDEs was used to optimize the model to be used for the high-performance simulation of large scale systems in which the movement of system boundaries is controlled by reaction-diffusion phenomena.

The investigated problem was the degradation of a magnesium block inside a solution, in which the surface of the block moves due to the reaction-diffusion phenomena in the metal-medium interface. The implemented model showed a good agreement with the experimental data in terms of the degradation rate and chemical reactions, and the parallel efficiency and linear scalability were appropriate in performance evaluation tests. For the next stage of the study, it could be interesting to evaluate the model and its performance on a much larger system and tune the resources and memory usage by testing different preconditioners and solvers.

## 7.A Comparing the performance of different combinations of KSP types and preconditioners

In order to yield the highest performance, a variety of different combinations of KSP types and preconditioners were evaluated, such as Conjugate Gradients (CG), Successive Over-Relaxation (SOR), block Jacobi, and Algebraic Multi-grid (AMG), and Generalized Minimal Residual Method (GMRES). The tests were performed using 6 MPI cores on an Ubuntu machine with an Intel Core i7-8850H CPU (2.6 GHz of clock speed) and a total available memory of 32 GB.

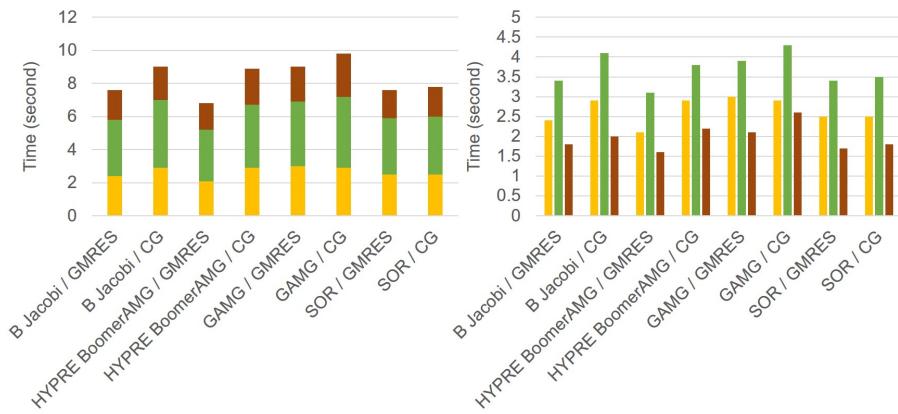


Figure 7.12: Performance test result for various combinations of preconditioners and solvers. Results are broken down into contributions for each PDE, which are plotted cumulatively and separately in the left and right plots, respectively.

Table 7.2: Performance test result for various combinations of preconditioners and solvers, presented by the execution time of each PDE.

		B Jacobi / GMRES	B Jacobi / CG	HYPRE / GMRES	HYPRE / CG	GAMG / GMRES	GAMG / CG	SOR / GMRES	SOR / CG
Solution time of each time step (s)	LS PDE	2.4	2.9	2.1	2.9	3.0	2.9	2.5	2.5
	Mg PDE	3.4	4.1	3.1	3.8	3.9	4.3	3.4	3.5
	Film PDE	1.8	2.0	1.6	2.2	2.1	2.6	1.7	1.8
Total time (s)		7.6	9.0	6.8	8.9	9.0	9.8	7.6	7.8

# Chapter 8

## BioDeg software

This chapter is based on previously published content in the *Journal of Open Source Software*:

M. Barzegari, and L. Geris, "BioDeg: A finite element software for the simulation of the corrosion and biodegradation process in metallic biomaterials," *Journal of Open Source Software*, vol. 7, p. 4281, 2022.

### 8.1 Summary

BioDeg is an open-source software written in FreeFEM (a domain-specific language for finite element programming), C++, and Python for modeling the degradation of metallic biomaterials and simulating the biodegradation behavior of medical devices and implants in corrosion experiments. The underlying mathematical and computational models are already validated in the previous contributions [102, 103] by comparing the predictions made by the code with the experimentally obtained quantities. BioDeg supports simulating the change of morphology of the biodegradable part, release of materials, formation of surface corrosion products, the effect of the surrounding environment (such as various electrolyte solutions and chemical components as well as the presence of fluid flow), and change of other quantities (such as pH). It features command-line and graphical interfaces, being available on all major operating systems and platforms. BioDeg is

designed with open standards in mind and as a result, can be easily integrated into other established workflows such as topology optimization and tissue growth models.

## 8.2 Statement of need

Biomaterials, the substances we put inside the body to replace/repair a lost functionality, can be classified into two categories from a biodegradation perspective: bio-inert and biodegradable. While the former type has a few interactions with its surrounding environment, the latter gradually disappears and gets absorbed by the body [1, 38]. Although bio-inert biomaterials show a great performance especially in fixation applications, they bring an important problem into play: they remain in the body forever or require additional surgery to remove them. Biodegradable materials do not have this problem, and in the case of metallic biomaterials, they also provide a suitable mechanical stability profile. However, taking advantage of them requires tuning the degradation parameters and material release rate [4].

This problem is usually investigated by conducting *in-vitro* (in the lab) and *in-vivo* (in a living system) tests of biodegradable metallic implants, which requires conducting multiple experiments for different scenarios and situations and spending a lot of resources. Developing a quantitative mathematical model of the degradation process is a proper solution to this issue by allowing researchers to study the biodegradation behavior of any desired implant *in-silico* (in the computer) prior to conducting any *in-vitro* or *in-vivo* experiments. Developed mathematical models can be simulated using efficient numerical methods such as the finite element method.

Although there is a wide variety of relevant theoretical studies in the field of corrosion and biodegradation simulation [42, 46, 44, 49, 75, 104, 105], to the best of the authors' knowledge, none of the developed models has been publicly released together with their code to be evaluated by the scientific community. It means that none of the previous contributions are available as a software tool for relevant users such as researchers in biomaterials science, regenerative medicine (tissue engineering), and biomedical engineering.

Moreover, the available tools used to implement aforementioned models are not open-source or have the issue of restricted access. For example, COMSOL

multiphysics (COMSOL Inc., USA) has a module for mechanistic corrosion simulation, although it lacks certain research-related features like simulating the change of morphology of the corroding object and freedom in defining the chemical reactions. Another example in this regard is the MuPhyS software [75, 106] developed by the research group of Electrochemical and Surface Engineering (SURF), Vrije Universiteit Brussel, Brussels, Belgium. Despite being able to perform mechanistic biodegradation simulations, MuPhys is a closed-source software for in-house applications and is not publicly available to general users.

BioDeg is a fully open-source biodegradation and corrosion simulation software, currently validated for magnesium biomaterials [102]. A cross-platform and easy-to-use user interface makes it possible for anyone to use the program to simulate the biodegradation behavior of metallic implants with any shape of interest without concerning the technical aspects of the code. Technically speaking, BioDeg uses the finite element method to solve a set of partial differential equations derived mechanistically from the chemistry of biodegradation and utilizes high-performance computing (HPC) techniques to make the computation scalable to hundreds and thousands of computational nodes.

## 8.3 Overview of BioDeg

### 8.3.1 Basic concepts

From a general point of view, material modeling techniques are categorized as either stochastic or deterministic. In stochastic modeling, material state evolution is modeled by a set of semi-random rules, implying the conditions in which materials interact with their surrounding environment. On the other hand, deterministic modeling takes advantage of conservative laws in physics, formulating the interactions of materials by means of the rules such as the ones coming from transport phenomena [107].

BioDeg works based on the principles of deterministic modeling for predicting the biodegradation and corrosion of metallic materials. Several studies have demonstrated that this type of modeling has a high potential in representing the biodegradation and corrosion phenomena [75, 49, 105]. BioDeg is built upon a mechanistic formulation of the biodegradation process [102]. The developed model captures the release of metallic ions, changes in pH, the

formation of a protective film, the dissolution of this film in presence of different ions, and the effect of fluid flow of the surrounding solution. This has been accomplished by deriving a system of time-dependent reaction-diffusion-convection partial differential equations (PDEs) from the underlying oxidation-reduction reactions. The level set formalism [55] was employed to track the biodegradation interface between the biomaterial and its surroundings, enabling the model to monitor the morphological changes of the investigated implant/device during the biodegradation process. As an example of BioDeg output, figure 8.1 demonstrates the simulation results of the biodegradation of an orthopedics screw, in which the white body and the color contour show the degrading shape of the screw and concentration of metallic ions as they are released over time, respectively. The current model is able to simulate diffusion-based corrosion only, and localized corrosion is not supported yet, but further development is being done to make the model capable of simulating non-uniform corrosion as well.

### 8.3.2 Implementation

Implementing a parallel 3D computational model of the developed mathematical framework allows studying the corrosion and biodegradation behavior of medical implants and devices *in-silico* no matter the complexity of the implant shape. This has been achieved by solving the derived system of equations using the finite element method on any arbitrary 3D mesh.

The computational model (BioDeg-core) is implemented in FreeFEM [61], a domain-specific language for PDE solving and finite element computations. While having a syntax almost identical to C++, FreeFEM provides rich interfaces to well-known scientific computing libraries for performing relevant tasks such as mesh generation, solving linear system of equations, various IO operations, and HPC. This fact makes FreeFEM a perfect choice for the implementation of complex mathematical models, especially for resource-demanding applications. Additionally, a general preprocessor for embedding desired 3D shapes inside a container (the medium/electrolyte in biodegradation simulations) is implemented in FreeFEM using the ParMmg parallel mesh manipulation library [108]. This helps non-technical users to perform simulations on their desired geometries. The preprocessor performs mesh refinement on the interface of the implant and the medium to increase the numerical accuracy of interface tracking. This mesh refinement is based on

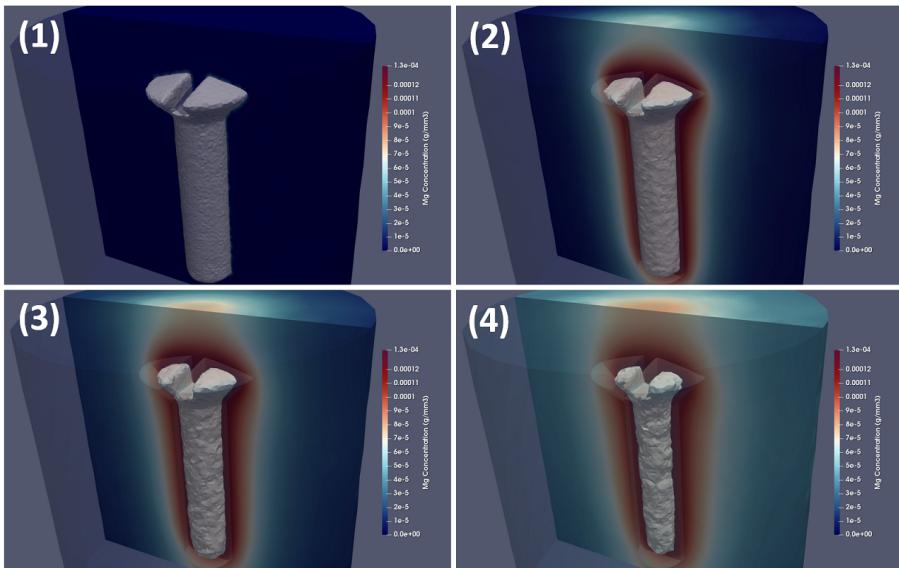


Figure 8.1: Simulation results of the degradation process of a screw made of magnesium inside a saline ( $\text{NaCl}$ ) solution, leading to a high rate of degradation. The contours display the concentration of magnesium ions on a cross-section view of the medium beside the moving surface of the screw. The screw was selected to be small (1 millimeter) so that the effect of degradation can be observed in a smaller time window. Each sub-figure shows a time difference of 24 hours. This example is provided along with the code.

the level-set function used to measure the distance of each element of the mesh to the biodegradable part surface. The preprocessor uses this function and a set of user-defined values (like the minimum and maximum element size) to refine the elements on the moving corrosion front. BioDeg supports tetrahedral elements only, which is a limitation rooted in supported element types in FreeFEM.

The user interface (BioDeg-UI) is implemented using C++ and Qt toolkit. A front-end using Qt and a back-end in FreeFEM enable BioDeg to be cross-platform and be able to run on major operating systems including Linux, Windows, and macOS. The postprocessing of the results is carried out by the open-source software ParaView [83], which is also a cross-platform postprocessor.

### 8.3.3 High-performance computing

Biodegradation simulations in BioDeg can become complex and resource-demanding. The reason lies within the refined mesh on the metal-electrolyte interface. This refined mesh is required to increase the accuracy of employed numerical schemes to track the moving corrosion front [102, 103]. This leads to an expensive computational model, and as a result, employing parallel computing and HPC techniques becomes crucial to make it possible to scale the simulations to hundreds or thousands of computational nodes and decrease the simulation execution time. To this end, a high-performance domain decomposition approach was employed to partition the mesh and distribute the workload to available computing resources [88]. Additionally, efficient preconditioners and solvers were used to solve the linear system of equations resulting from the finite element discretization. This was achieved by using the rich PETSc library [64], which provides a collection of efficient preconditioners and solvers for solving PDEs.

### 8.3.4 Installation

The main prerequisites of BioDeg are FreeFEM, PETSc, Mmg, and an MPI runtime (such as OpenMPI or MPICH). The PETSc library contains the rest of the dependencies. Since FreeFEM provides appropriate interfaces to Mmg and PETSc and includes installation scripts for building and installing them, all the requirements can be met by building/installing FreeFEM with the PETSc link enabled. While a set of pre-built binaries are available for download on the FreeFEM website, the user manual of BioDeg includes a section for instructing the users to build FreeFEM to achieve maximum flexibility on their systems.

The next step is to build or install the BioDeg UI, the graphical user interface of BioDeg. Stable binaries can be downloaded from the Release section of the GitHub repository for Windows and Linux, but if the user prefers to compile it from source code, the build process is straightforward as well. The prerequisites for building the UI are Qt framework and CMake, which need to be installed for the target operating system. The rest of the process is a standard CMake build routine, which can be followed from the installation instruction in the BioDeg User Manual or the GitHub repository. After building/installing BioDeg UI, the user can run it to setup and execute simulations.

In case the user does not need to work with the UI, like in HPC environments or for advanced users, the previous steps for building BioDeg-UI can be skipped, and the user can directly run the core model by executing `mpiexec -n N FreeFem++-mpi core/src/main.edp -v 0` to parallelize the computation into N MPI processes. In this way, the simulation can be configured by dozens of available command-line arguments, documented in the BioDeg User Manual.

For graphical output, the postprocessing of the simulation results is handled by ParaView, so the users should have it installed on the system. Stable binaries for various platforms can be found on the ParaView website, including portable versions which do not require installation and are ready to run immediately after download. BioDeg UI can help visualize the results by executing specific filtering processes in ParaView, which can also be done manually by following the postprocessing guideline provided in the BioDeg User Manual.

### 8.3.5 Documentation

A comprehensive user manual is provided for user-oriented documentation on installing the software, preparing the simulations, running the models, and postprocessing the results. As this is being developed in a separate repository (BioDeg-doc [109]), additional documents and supporting materials will be added eventually.

Since FreeFEM does not support developer-focused technical documentation (like Doxygen), the core code is fully commented for easier contribution. Further technical details regarding the development, like the theoretical foundations of the computational models, has been added to the BioDeg-doc repository [109]. Both the user manual and the theory guide are accessible from the Help menu on the BioDeg UI.

### 8.3.6 Contribution

BioDeg is developed using a wide range of open-source tools and is released under GPLv3, so unconstrained use and reuse are encouraged and welcome. The code has been initially developed as part of a PhD research project, but further contributions, either on the development of the core models or providing more supporting materials, are highly welcome. More guidelines for such contributions can be found in the CONTRIBUTING.md file in the

GitHub repository. Previous contributions are acknowledged in the BioDeg User Manual.

### 8.3.7 Published works

The code has been already used in publishing 2 scientific works so far, and there are several studies ongoing to be prepared for publication. In the published works, BioDeg was employed to simulate the behavior of high-pure magnesium (HP Mg) in immersion corrosion tests [102] as well as to evaluate the efficiency and scalability of the employed parallelization schemes for reaction-diffusion systems with moving boundaries [103]. Currently, the code is being used for simulating the degradation of magnesium in hydrodynamics (perfusion) conditions, in which the underlying phenomena are investigated in the presence of fluid flow. Moreover, BioDeg is being coupled with a topology optimization code for optimizing the shape of biodegradable structures for medical applications. Later, it will be integrated it with an optimization code and bone tissue growth models for improving the design process of biodegradable orthopedics implants.

## 8.4 Acknowledgements

The research in which the code is developed is financially supported by the Prosperos project, funded by the Interreg VA Flanders The Netherlands program, CCI grant no. 2014TC16RFCB046 and by the Fund for Scientific Research Flanders (FWO), grant G085018N. We also acknowledge support from the European Research Council under the European Union's Horizon 2020 research and innovation programme, ERC CoG 772418.

# Chapter 9

## Bayesian parameter estimation of the computational models

This chapter is based on previously published content in the *Journal of Open Source Education*:

M. Barzegari, and L. Geris, "An open source crash course on parameter estimation of computational models using a Bayesian optimization approach," *Journal of Open Source Education*, vol. 4, p. 89, 2021.

This publication included didactical and educational materials in Jupyter Notebooks format to make it possible for readers to follow the principles interactively. The introductory part of the Jupyter Notebook file is included in this chapter as well.

### 9.1 Summary

Parameter estimation is a crucial aspect of computational modeling projects, especially the ones that deal with ordinary differential equations (ODE) or partial differential equation (PDE) models. Well-known examples in this regard are models derived from a basic balance or conservation law, such

as mass balance or heat transfer problems. For real-world applications, these equations contain some coefficients that cannot be obtained directly from published scientific materials or experimental studies [110]. One of the best solutions to this challenge is constructing an inverse problem.

According to [111], inverse modeling is the use of the results of some measurements of observable parameters to infer the values of the model parameters. Put differently, what we want to do is estimate parameters that cannot be directly measured for our computational model. This is also called parameter estimation or model calibration [112]. Indeed, we calibrate our model to act similarly to available experimental data, and then this calibrated model can be used to simulate other scenarios that haven't been tested yet in the experiments. This is a common process in a lot of modeling problems in science and engineering.

Take a simple reaction-diffusion equation as an example, in which the change of the concentration of a sample chemical component  $C$  is studied over time. By assuming that the correlated chemical reaction is  $A + 2B \rightleftharpoons C$ , occurring in a diffusible medium (such as a chemical solution), the PDE to describe the mass transfer phenomenon over time can be written as [53]:

$$\frac{\partial[C]}{\partial t} = \nabla \cdot (D_C \nabla [C]) + k_1[A][B]^2 - k_2[C] \quad (9.1)$$

in which  $[X]$  denotes the concentration of the chemical component  $X$ , the  $D_C$  is the diffusion coefficient of  $C$  in the medium, and  $k_1$  and  $k_2$  are the rates of the forward and backward reactions, respectively. To solve this PDE numerically and get quantitative data (the goal of most of the scientific computing projects), we need to know the value of  $D_C$ ,  $k_1$ , and  $k_2$ , which is usually hard-to-find in the literature.

As mentioned above, one solution is to solve the inverse problem, in which we can use optimization techniques to minimize the difference between the model output and experimental data. Bayesian optimization is one of the most efficient approaches in this regard [68]. HyperOpt [113] is a Python package that provides easy-to-use interfaces to implement a Bayesian optimization problem, making it a good choice for both educational and practical purposes. In our educational module, we used this package to teach the principles of an efficient parameter estimation pipeline.

For demonstration purposes, an interpolation problem is solved by using the

parameter estimation techniques that a computational modeling researcher employs for model calibration. Indeed, the computationally intensive code is replaced with a simple function evaluator, which helps students to learn the core concepts without waiting too much for the process to finish. Students will be guided through several steps of refining the results inside the notebook, where the interactive computing environment of Jupyter facilitates exploring the implementation more efficiently in comparison to traditional educational materials.

## 9.2 Statement of Need

Despite its simplicity, building an inverse problem is hard for many students. The problem is that, although it is relatively simple to describe the process visually, implementing it for a practical application becomes challenging in its early stages. In this educational module, a simple optimization problem is implemented in a Jupyter notebook to teach students how to construct an inverse problem and tune it to get better results in such a problem. In this way, students can work on a real-world optimization problem in an interactive environment and learn the concepts behind taking advantage of a modern optimization method (Bayesian approach) for parameter estimation of a computational model.

Our notebook is a modern learning module for relatively old and frequently-used concepts (global optimization, Bayesian techniques, inverse problems). It has been designed to be useful for both teachers and students. Students can use it as a self-study guide for parameter estimation and inverse problem construction, while teachers can change the underlying problem to any other desired one easily and make the learning module compatible with their own teaching requirements.

## 9.3 Learning objectives

Upon completion, students will be able to:

- Understand the concept and necessity of parameter estimation in science and engineering

- Describe what the whole process of Bayesian optimization is all about
- Define and implement a Bayesian optimization workflow for parameter estimation of common use-cases
- Critically evaluate the output of the process and fine-tune the setup of the Bayesian optimization
- Apply the obtained knowledge to any kind of models that are commonly used in science and engineering

## 9.4 Prerequisites

In order to go through the learning module, the students should have a working knowledge of programming in Python. Additionally, a basic understanding of mathematics is required to get the concept of models in science and engineering. The given example is a mathematical model derived from differential equations, so knowledge of differential equations can help to understand the importance of parameter estimation in these widely-used models. However, in case of necessity, the example can be replaced by any other relevant one for the target learners.

## 9.5 Pedagogy and instructional design

The provided material is in the format of a crash course, which is suitable for being taught in one session of undergraduate or graduate courses for science and engineering students. Courses to which this material is relevant can be “optimization”, “scientific computing”, or “parametric design”. The material may also be useful for relevant educational projects for the target students, in which they can employ the learned techniques to construct efficient inverse problems for parameter estimation.

The teaching strategy is based on the worked-example effect [114], in which an example of parameter estimation is fully implemented to allow students to play with and modify the code to have their own reflection in class discussions. Basic prior knowledge of Python suffices as the problem doesn't involve students with complicated programming stuff. The student-centric

characteristic of this crash course helps teachers to adopt the material easily and integrate it into an existing syllabus of relevant courses in science and engineering.

## 9.6 Getting started

The learning material is provided as a single Jupyter notebook, in which all the steps of constructing an inverse problem are described in detail with accompanying Python codes. A very simple simulation code (in the context of an interpolation problem) is also provided and can be found in the repository. The code inside the notebook calls this external program at certain points to mimic the interaction of the parameter estimation routine and the main computational code that contains the unknown parameters.

To get started with the module, the user should set up the environment first. The setup instructions are provided in the `README.md` file of the repository. After setting up the Jupyter notebook and installing the required packages, the user can navigate to the `src` folder and run the notebook file. No further action is needed as the content of the notebook is self-explanatory and easy-to-follow.

## 9.7 Acknowledgements

This research is financially supported by the Prosperos project, funded by the Interreg VA Flanders The Netherlands program, CCI grant no. 2014TC16RFCB046 and by the Fund for Scientific Research Flanders (FWO), grant G085018N. We acknowledge support from the European Research Council under the European Union's Horizon 2020 research and innovation programmes, ERC CoG 772418.

## 9.A A glimpse of the Jupyter notebook

### 9.A.1 Introduction

As the name implies, the parameter estimation process deals with approximating unknown parameters, the factors that define a system or its operation. In science and engineering, this can be seen as a sub-category of optimization techniques since we seek to find the optimal state of a system. After finding the desired state, we look for the parameters contributing to such a state. It is indeed what parameter estimation is all about. Seems complex? Look at it as a calibration process, in which a machine, tool, or system is tuned to produce a correct output. Imagine you want to calibrate a machine with 3 knobs. How do you do the calibration? You compare the output with a reference, something you know the machine should produce, and then try to adjust the knobs such that the output matches the reference. It is how calibration works, no? Take a thermometer as an example. You have a reference temperature, like boiling water at 100 degrees Celsius, and 3 knobs on the device. You continue turning the knobs to see 100 appearing on the machine, and by doing that, you calibrate the thermometer. In this way, you have estimated the unknown parameters (the 3 knobs) of the device. After being calibrated, you can use the thermometer to measure any temperature.

Now, instead of the machine, assume you want to perform the same process on an engineering system. Each system (or, let's say, model) has a certain number of parameters to be tuned. After calibrating the system (model) with the reference data (a data we already know is correct), we can assure that the system's output is more or less valid if being used for another measurement (prediction). Real-world systems in science and engineering contain some parameters that cannot be obtained directly from published scientific materials or experimental studies. Thus, we should estimate them using a calibration (optimization) strategy, a process that is generally called an inverse problem.

An inverse problem in science is usually referred to the process of calculating the causal factors from a set of observations that have produced them. Inverse problems are important because they tell us about parameters that we cannot directly observe. That's exactly what we want to do: estimating parameters that we cannot directly measure for our model. Indeed, we calibrate our model to act similarly to available experimental data, and then this calibrated model can be used to simulate other scenarios that we have not tested in experiments. This

concept is similar to the training process of machine learning models. You train the model by making it fit to previously available data and asking it to predict unseen data.

To construct an inverse problem for computational models, we can take advantage of conventional optimization approaches. The goal is to minimize or maximize a function, or more technically speaking, an objective function. Mathematical optimization is selecting the best element (concerning some criterion) from some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing an objective function by systematically choosing input values from an allowed set and computing the output of the system. More generally, optimization is the process of finding the optimum value of an objective function given a defined domain (or input). To wrap up, the essential concepts here are the objective function (what we want to minimize) and the domain space (values of the parameters over which we minimize the objective).

Back to our example problem, our objective function will be the difference between the produced output of our simulation and the experimental data of the exact condition, which is also called "loss". In other words, we change the coefficients such that the simulation output would be the same as (or close to) the experimental data that we already have. To this end, we can choose random values out of the domain space (the range that we search for appropriate values), evaluate the model with those values, and continue this process till we find the lowest loss possible. This can be a good approach as long as the cost function evaluation is cheap, which means the simulations run fast (because we need to run the simulation to evaluate the cost function). The problem is, for most of the real-world models, running each simulation takes quite long. As a result, each iteration of the optimization algorithm is not cheap anymore. To overcome this issue, we use a Bayesian optimization strategy, a method that is usually employed to optimize expensive-to-evaluate functions.

## 9.A.2 Bayesian optimization

To describe how the Bayesian optimization approach helps us to overcome the problem mentioned above, I use the great description made by @WillKoehrsen. I can't explain it better (you can find the full interactive document at <https://github.com/WillKoehrsen/hyperparameter-optimization>:

Evaluating the objective function is the expensive part of optimization, so ideally, we want to limit calls to this function. One way we can limit calls is by choosing the next values to try in the objective function based on the past results. Bayesian optimization differs from random or grid search by doing exactly this: rather than just selecting from a grid uninformed by past objective function evaluations, Bayesian methods take into account the previous results to try more promising values. They work by constructing a probability model of the objective function (called a surrogate function)  $p(\text{score}|\text{parameters})$  which is much easier to optimize than the actual objective function.  $p(A | B)$ , the conditional probability, is the probability of A given B, i.e., A after B is observed.

After each evaluation of the objective function, the algorithm updates the probability model (usually given as  $p(y|x)$ ) incorporating the new results. Sequential Model-Based Optimization (SMBO) methods are a formalization of Bayesian optimization that updates the probability model sequentially: every evaluation of the objective function with a set of values updates the model with the idea that eventually the model will come to represent the true objective function. This is an application of Bayesian Reasoning. The algorithm forms an initial idea of the objective function and updates it with each new piece of evidence.

The next values to try in the objective function are selected by the algorithm optimizing the probability model (surrogate function), usually with a criterion known as Expected Improvement. Finding the values that will yield the greatest expected improvement in the surrogate function is much cheaper than evaluating the objective function itself. By choosing the next values based on a model rather than randomly, we hope that the algorithm would converge to the true best values much quicker. The overall goal is to evaluate the objective function fewer times by spending a little more time choosing the next values. Overall, Bayesian Optimization and SMBO methods:

- Converge to a lower score of the objective function than random search
- Require far less time to find the optimum of the objective function

So, we get both a faster optimization and a better result. These are both two desirable outcomes, especially when we are working with heavy computational models!

According to this simplified description, Bayesian optimization is a great candidate to perform the parameter estimation of PDE-based computational models. If you are interested to know more about the mathematical aspects of the Bayesian optimization, you may have a look at the SigOpt Bayesian Optimization Primer.

In this notebook, we implement the whole process of a Bayesian optimization strategy, including constructing a cost function by calling the simulation code, performing the optimization, and postprocessing the results. To do this, we use Python and HyperOpt, an open-source Python library for Bayesian optimization that implements SMBO using the Tree-structured Parzen Estimator (TPE). TPE, along with Gaussian Processes and Random Forest Regression, are the algorithms that can be used in the SMBO method to construct the probability model (surrogate function). We don't need to worry about implementing the algorithm because Hyperopt takes care of that for us. We have to make sure we have correctly defined the objective function and the domain of values to search over.

### 9.A.3 Sample problem

Instead of a PDE model, we use a more straightforward problem to focus more on the optimization rather than the numerical simulation of the PDE. The problem with which the optimization algorithm interact is fitting a 4th-order polynomial equation on some experimental data. You should notice that the optimization algorithm is entirely unaware of the fitting problem behind the objective function.

The objective function is implemented in a general way: calling the external simulation code, gathering produced output, and computing the loss. To make this as real-world as possible, I implemented an external Python code that takes polynomial coefficients and computes the function values on desired points. This data is saved on the disk and then retrieved by the objective function of the main optimization code to calculate the loss. In a real-world application, the external simulation code takes the coefficients (such as the reaction rates in the above PDE example), performs the simulation, and writes the output to the disk. The rest of the process is the same. In the next step, the optimization algorithm changes the parameters and calls the external simulation code again to see the loss of the new parameters. This process continues iteratively to a certain number of iterations.

The experimental data, which are indeed 21 points of a 4th-order polynomial function in the range  $[0, 5]$ , are stored in a CSV file. Each line contains one point, better to say a pair of two values defining a point, and this exactly what it can be in a real scenario. For example, it can contain the value of a chemical

component concentration over time, in which the first and second values of each line would be time and concentration, respectively.

The optimization output of this sample problem will be similar to Fig. 9.1 .

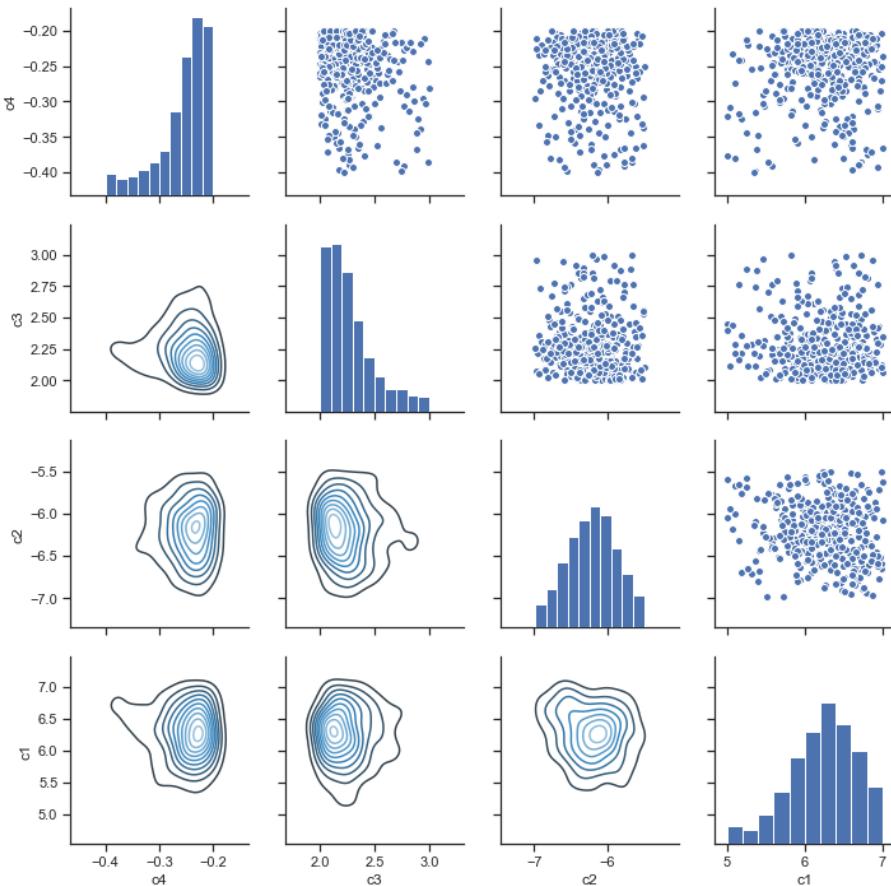


Figure 9.1: A typical output of the parameter estimation process, plotted using the Seaborn Python package, which shows how the optimization algorithm has chosen the values of different parameters in different iterations by comparing the distribution of points 2 by 2 for 4 unknown parameters of the polynomial model.

## **Chapter 10**

### **Applications: Mandible plate**



## **Chapter 11**

# **Applications: Acetabular cup**



## **Chapter 12**

# **Applications: Mechanical integrity of infilled structures**



# **Chapter 13**

## **Conclusion**

...



# Bibliography

- [1] Y. Zheng, X. Gu, and F. Witte, "Biodegradable metals," *Materials Science and Engineering: R: Reports*, vol. 77, pp. 1–34, mar 2014.
- [2] Y. Liu, Y. Zheng, X.-H. Chen, J.-A. Yang, H. Pan, D. Chen, L. Wang, J. Zhang, D. Zhu, S. Wu, K. W. K. Yeung, R.-C. Zeng, Y. Han, and S. Guan, "Fundamental theory of biodegradable metals—definition, criteria, and design," *Advanced Functional Materials*, vol. 29, p. 1805402, feb 2019.
- [3] H.-S. Han, S. Loffredo, I. Jun, J. Edwards, Y.-C. Kim, H.-K. Seok, F. Witte, D. Mantovani, and S. Glyn-Jones, "Current status and outlook on the clinical translation of biodegradable metals," *Materials Today*, vol. 23, pp. 57–71, mar 2019.
- [4] D. Zhao, F. Witte, F. Lu, J. Wang, J. Li, and L. Qin, "Current status on clinical applications of magnesium-based orthopaedic implants: A review from clinical translational perspective," *Biomaterials*, vol. 112, pp. 287–302, jan 2017.
- [5] Z. ZHEN, T. fei XI, and Y. feng ZHENG, "A review on in vitro corrosion performance test of biodegradable metallic materials," *Transactions of Nonferrous Metals Society of China*, vol. 23, pp. 2283–2293, aug 2013.
- [6] R. Willumeit-Römer, "The interface between degradable mg and tissue," *JOM*, vol. 71, pp. 1447–1455, feb 2019.
- [7] J. Venezuela and M. Dargusch, "The influence of alloying and fabrication techniques on the mechanical properties, biodegradability and biocompatibility of zinc: A comprehensive review," *Acta Biomaterialia*, vol. 87, pp. 1–40, mar 2019.

- [8] E. Mostaed, M. Sikora-Jasinska, J. W. Drelich, and M. Vedani, "Zinc-based alloys for degradable vascular stent applications," *Acta Biomaterialia*, vol. 71, pp. 1–23, apr 2018.
- [9] M. Schinhammer, A. C. Hänzi, J. F. Löffler, and P. J. Uggowitzer, "Design strategy for biodegradable fe-based alloys for medical applications," *Acta Biomaterialia*, vol. 6, pp. 1705–1713, may 2010.
- [10] M. Esmaily, J. Svensson, S. Fajardo, N. Birbilis, G. Frankel, S. Virtanen, R. Arrabal, S. Thomas, and L. Johansson, "Fundamentals and advances in magnesium alloy corrosion," *Progress in Materials Science*, vol. 89, pp. 92–193, aug 2017.
- [11] B. Heublein, "Biocorrosion of magnesium alloys: a new principle in cardiovascular implant technology?," *Heart*, vol. 89, pp. 651–656, jun 2003.
- [12] M. P. Staiger, A. M. Pietak, J. Huadmai, and G. Dias, "Magnesium and its alloys as orthopedic biomaterials: A review," *Biomaterials*, vol. 27, pp. 1728–1734, mar 2006.
- [13] J. Walker, S. Shadanbaz, T. B. F. Woodfield, M. P. Staiger, and G. J. Dias, "Magnesium biomaterials for orthopedic application: A review from a biological perspective," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 102, pp. 1316–1331, jan 2014.
- [14] F. Cecchinato, N. A. Agha, A. H. Martinez-Sanchez, B. J. C. Luthringer, F. Feyerabend, R. Jimbo, R. Willumeit-Römer, and A. Wennerberg, "Influence of magnesium alloy degradation on undifferentiated human cells," *PLOS ONE*, vol. 10, p. e0142117, nov 2015.
- [15] D. Mei, S. V. Lamaka, X. Lu, and M. L. Zheludkevich, "Selecting medium for corrosion testing of bioabsorbable magnesium and other metals – a critical review," *Corrosion Science*, vol. 171, p. 108722, jul 2020.
- [16] D. Mei, S. V. Lamaka, J. Gonzalez, F. Feyerabend, R. Willumeit-Römer, and M. L. Zheludkevich, "The role of individual components of simulated body fluid on the corrosion behavior of commercially pure mg," *Corrosion Science*, vol. 147, pp. 81–93, feb 2019.
- [17] R.-C. Zeng, Y. Hu, S.-K. Guan, H.-Z. Cui, and E.-H. Han, "Corrosion of magnesium alloy AZ31: The influence of bicarbonate, sulphate,

- hydrogen phosphate and dihydrogen phosphate ions in saline solution," *Corrosion Science*, vol. 86, pp. 171–182, sep 2014.
- [18] S. Johnston, M. Dargusch, and A. Atrens, "Building towards a standardised approach to biocorrosion studies: a review of factors influencing mg corrosion in vitro pertinent to in vivo corrosion," *Science China Materials*, vol. 61, pp. 475–500, dec 2017.
- [19] S. V. Lamaka, J. Gonzalez, D. Mei, F. Feyerabend, R. Willumeit-Römer, and M. L. Zheludkevich, "Local pH and its evolution near mg alloy surfaces exposed to simulated body fluids," *Advanced Materials Interfaces*, vol. 5, p. 1800169, jun 2018.
- [20] D. Mei, S. V. Lamaka, C. Feiler, and M. L. Zheludkevich, "The effect of small-molecule bio-relevant organic components at low concentration on the corrosion of commercially pure mg and mg-0.8ca alloy: An overall perspective," *Corrosion Science*, vol. 153, pp. 258–271, jun 2019.
- [21] B. Hadzima, M. Mhaede, and F. Pastorek, "Electrochemical characteristics of calcium-phosphatized AZ31 magnesium alloy in 0.9 % NaCl solution," *Journal of Materials Science: Materials in Medicine*, vol. 25, pp. 1227–1237, jan 2014.
- [22] X. Lu, Y. Li, P. Ju, Y. Chen, J. Yang, K. Qian, T. Zhang, and F. Wang, "Unveiling the inhibition mechanism of an effective inhibitor for AZ91 mg alloy," *Corrosion Science*, vol. 148, pp. 264–271, mar 2019.
- [23] Y. Li, X. Lu, K. Wu, L. Yang, T. Zhang, and F. Wang, "Exploration the inhibition mechanism of sodium dodecyl sulfate on mg alloy," *Corrosion Science*, vol. 168, p. 108559, may 2020.
- [24] A. Atrens, G.-L. Song, M. Liu, Z. Shi, F. Cao, and M. S. Dargusch, "Review of recent developments in the field of magnesium corrosion," *Advanced Engineering Materials*, vol. 17, pp. 400–453, jan 2015.
- [25] C. Schille, M. Braun, H. Wendel, L. Scheideler, N. Hort, H.-P. Reichel, E. Schweizer, and J. Geis-Gerstorfer, "Corrosion of experimental magnesium alloys in blood and PBS: A gravimetric and microscopic evaluation," *Materials Science and Engineering: B*, vol. 176, pp. 1797–1801, dec 2011.

- [26] D. Xue, Y. Yun, Z. Tan, Z. Dong, and M. J. Schulz, "In vivo and in vitro degradation behavior of magnesium alloys as biomaterials," *Journal of Materials Science & Technology*, vol. 28, pp. 261–267, mar 2012.
- [27] Y. Xin, T. Hu, and P. K. Chu, "Degradation behaviour of pure magnesium in simulated body fluids with different concentrations of  $\text{HCO}_3$ ," *Corrosion Science*, vol. 53, pp. 1522–1528, apr 2011.
- [28] N. A. Agha, F. Feyerabend, B. Mihailova, S. Heidrich, U. Bismayer, and R. Willumeit-Römer, "Magnesium degradation influenced by buffering salts in concentrations typical of in vitro and in vivo models," *Materials Science and Engineering: C*, vol. 58, pp. 817–825, jan 2016.
- [29] L.-Y. Cui, Y. Hu, R.-C. Zeng, Y.-X. Yang, D.-D. Sun, S.-Q. Li, F. Zhang, and E.-H. Han, "New insights into the effect of tris-HCl and tris on corrosion of magnesium alloy in presence of bicarbonate, sulfate, hydrogen phosphate and dihydrogen phosphate ions," *Journal of Materials Science & Technology*, vol. 33, pp. 971–986, sep 2017.
- [30] M. B. Kannan, H. Khakbaz, and A. Yamamoto, "Understanding the influence of HEPES buffer concentration on the biodegradation of pure magnesium: An electrochemical study," *Materials Chemistry and Physics*, vol. 197, pp. 47–56, aug 2017.
- [31] Y. Song, D. Shan, R. Chen, F. Zhang, and E.-H. Han, "Biodegradable behaviors of AZ31 magnesium alloy in simulated body fluid," *Materials Science and Engineering: C*, vol. 29, pp. 1039–1045, apr 2009.
- [32] E. L. Silva, S. V. Lamaka, D. Mei, and M. L. Zheludkevich, "The reduction of dissolved oxygen during magnesium corrosion," *ChemistryOpen*, vol. 7, pp. 664–668, aug 2018.
- [33] P. Jiang, C. Blawert, N. Scharnagl, and M. L. Zheludkevich, "Influence of water purity on the corrosion behavior of  $\text{Mg}_{0.5}\text{Zn}_x$  ( $x=\text{Ca}, \text{Ge}$ ) alloys," *Corrosion Science*, vol. 153, pp. 62–73, jun 2019.
- [34] D. Mei, C. Wang, S. V. Lamaka, and M. L. Zheludkevich, "Clarifying the influence of albumin on the initial stages of magnesium corrosion in Hank's balanced salt solution," *Journal of Magnesium and Alloys*, vol. 9, pp. 805–817, may 2021.

- [35] S. Johnston, Z. Shi, and A. Atrens, "The influence of pH on the corrosion rate of high-purity mg, AZ91 and ZE41 in bicarbonate buffered hanks' solution," *Corrosion Science*, vol. 101, pp. 182–192, dec 2015.
- [36] S. Johnston, Z. Shi, J. Venezuela, C. Wen, M. S. Dargusch, and A. Atrens, "Investigating mg biocorrosion in vitro: Lessons learned and recommendations," *JOM*, vol. 71, pp. 1406–1413, jan 2019.
- [37] C. Wang, D. Mei, G. Wiese, L. Wang, M. Deng, S. V. Lamaka, and M. L. Zheludkevich, "High rate oxygen reduction reaction during corrosion of ultra-high-purity magnesium," *npj Materials Degradation*, vol. 4, dec 2020.
- [38] Y. Chen, Z. Xu, C. Smith, and J. Sankar, "Recent advances on the development of magnesium alloys for biodegradable implants," *Acta Biomaterialia*, vol. 10, pp. 4561–4573, nov 2014.
- [39] Y. Qin, P. Wen, H. Guo, D. Xia, Y. Zheng, L. Jauer, R. Poprawe, M. Voshage, and J. H. Schleifenbaum, "Additive manufacturing of biodegradable metals: Current research status and future perspectives," *Acta Biomaterialia*, vol. 98, pp. 3 – 22, 2019.
- [40] U. Riaz, I. Shabib, and W. Haider, "The current trends of mg alloys in biomedical applications—a review," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, dec 2018.
- [41] Y. Xin, K. Huo, H. Tao, G. Tang, and P. K. Chu, "Influence of aggressive ions on the degradation behavior of biomedical magnesium alloy in physiological environment," *Acta Biomaterialia*, vol. 4, pp. 2008–2015, nov 2008.
- [42] D. Gastaldi, V. Sassi, L. Petrini, M. Vedani, S. Trasatti, and F. Migliavacca, "Continuum damage model for bioresorbable magnesium alloy devices — application to coronary stents," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 4, pp. 352–365, apr 2011.
- [43] S. Ahmed, J. Ward, and Y. Liu, "Numerical modelling of effects of biphasic layers of corrosion products to the degradation of magnesium metal in vitro," *Materials*, vol. 11, p. 1, dec 2017.
- [44] J. Grogan, S. Leen, and P. McHugh, "A physical corrosion model for bioabsorbable metal stents," *Acta Biomaterialia*, vol. 10, pp. 2313–2322, may 2014.

- [45] Z. Shen, M. Zhao, D. Bian, D. Shen, X. Zhou, J. Liu, Y. Liu, H. Guo, and Y. Zheng, "Predicting the degradation behavior of magnesium alloys with a diffusion-based theoretical model and in vitro corrosion testing," *Journal of Materials Science & Technology*, vol. 35, pp. 1393–1402, jul 2019.
- [46] J. W. Wilder, C. Clemons, D. Golovaty, K. L. Kreider, G. W. Young, and R. S. Lillard, "An adaptive level set approach for modeling damage due to galvanic corrosion," *Journal of Engineering Mathematics*, vol. 91, pp. 121–142, nov 2014.
- [47] A.-K. Gartzke, S. Julmi, C. Klose, A.-C. Waselau, A. Meyer-Lindenberg, H. J. Maier, S. Besdo, and P. Wriggers, "A simulation model for the degradation of magnesium-based bone implants," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 101, p. 103411, jan 2020.
- [48] W. Sun, G. Liu, L. Wang, T. Wu, and Y. Liu, "An arbitrary lagrangian-eulerian model for studying the influences of corrosion product deposition on bimetallic corrosion," *Journal of Solid State Electrochemistry*, vol. 17, pp. 829–840, nov 2012.
- [49] P. Bajger, J. M. A. Ashbourn, V. Manhas, Y. Guyot, K. Lietaert, and L. Geris, "Mathematical modelling of the degradation behaviour of biodegradable metals," *Biomechanics and Modeling in Mechanobiology*, vol. 16, pp. 227–238, aug 2016.
- [50] J. Sanz-Herrera, E. Reina-Romo, and A. Boccaccini, "In silico design of magnesium implants: Macroscopic modeling," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 79, pp. 181–188, mar 2018.
- [51] Y. Gao, L. Wang, X. Gu, Z. Chu, M. Guo, and Y. Fan, "A quantitative study on magnesium alloy stent biodegradation," *Journal of Biomechanics*, vol. 74, pp. 98–105, jun 2018.
- [52] Y. Wang, J. Pan, X. Han, C. Sinka, and L. Ding, "A phenomenological model for the degradation of biodegradable polymers," *Biomaterials*, vol. 29, no. 23, pp. 3393–3401, 2008.
- [53] P. Grindrod, *The theory and applications of reaction-diffusion equations : patterns and waves*. Oxford University Press, 1996.
- [54] J. Crank, *Free and Moving Boundary Problems*. OUP Oxford, 1987.

- [55] S. O. Ronald Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer New York, 2002.
- [56] M. Strebl, M. Bruns, and S. Virtanen, “Editors’ choice—respirometric *in situ* methods for real-time monitoring of corrosion rates: Part i. atmospheric corrosion,” *Journal of The Electrochemical Society*, vol. 167, p. 021510, jan 2020.
- [57] P. Grathwohl, *Diffusion in Natural Porous Media: Contaminant Transport, Sorption/Desorption and Dissolution Kinetics*. Springer US, 1998.
- [58] D. Höche, “Simulation of corrosion product deposit layer growth on bare magnesium galvanically coupled to aluminum,” *Journal of The Electrochemical Society*, vol. 162, pp. C1–C11, nov 2014.
- [59] M. Barzegari and L. Geris, “Highly scalable numerical simulation of coupled reaction-diffusion systems with moving interfaces,”
- [60] S. Scheiner and C. Hellmich, “Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary,” *Corrosion Science*, vol. 49, pp. 319–346, feb 2007.
- [61] F. Hecht, “New development in freefem++,” *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012.
- [62] R. D. Falgout and U. M. Yang, “hypre: A library of high performance preconditioners,” in *Lecture Notes in Computer Science*, pp. 632–641, Springer Berlin Heidelberg, 2002.
- [63] Y. Saad and M. H. Schultz, “Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.
- [64] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, “PETSc Web page.” <https://www.mcs.anl.gov/petsc>, 2019.
- [65] P. Jolivet, F. Hecht, F. Nataf, and C. Prudhomme, “Scalable domain decomposition preconditioners for heterogeneous elliptic problems,” in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC 13, (New York, NY, USA), Association for Computing Machinery, 2013.

- [66] J. Schöberl, "NETGEN an advancing front 2d/3d-mesh generator based on abstract rules," *Computing and Visualization in Science*, vol. 1, pp. 41–52, jul 1997.
- [67] A. Ribes and C. Caremoli, "Salome platform component model for numerical simulation," in *31st Annual International Computer Software and Applications Conference - Vol. 2 - (COMPSAC 2007)*, IEEE, jul 2007.
- [68] J. Mockus, *Bayesian Approach to Global Optimization*. Springer Netherlands, 1989.
- [69] M. Mehrian, Y. Guyot, I. Papantoniou, S. Olofsson, M. Sonnaert, R. Misener, and L. Geris, "Maximizing neotissue growth kinetics in a perfusion bioreactor: An in silico strategy using model reduction and bayesian optimization," *Biotechnology and Bioengineering*, vol. 115, pp. 617–629, dec 2017.
- [70] S. H. Lee and J. C. Rasaiah, "Proton transfer and the mobilities of the H<sup>+</sup> and OH<sup>-</sup> ions from studies of a dissociating model for water," *The Journal of Chemical Physics*, vol. 135, p. 124505, sep 2011.
- [71] J. S. H. Graham C. Hill, *Chemistry in Context - Laboratory Manual*. Oxford University Press, 2001.
- [72] M. Abdalla, A. Joplin, M. Elahinia, and H. Ibrahim, "Corrosion modeling of magnesium and its alloys for biomedical applications: Review," *Corrosion and Materials Degradation*, vol. 1, pp. 219–248, jul 2020.
- [73] R. J. Santucci, M. E. McMahon, and J. R. Scully, "Utilization of chemical stability diagrams for improved understanding of electrochemical systems: evolution of solution chemistry towards equilibrium," *npj Materials Degradation*, vol. 2, jan 2018.
- [74] K. B. Deshpande, "Numerical modeling of micro-galvanic corrosion," *Electrochimica Acta*, vol. 56, pp. 1737–1745, jan 2011.
- [75] O. Dolgikh, H. Simillion, S. V. Lamaka, A. C. Bastos, H. B. Xue, M. G. Taryba, A. R. Oliveira, C. Allély, B. V. D. Bossche, K. V. D. Bergh, J. D. Strycker, and J. Deconinck, "Corrosion protection of steel cut-edges by hot-dip galvanized al(zn,mg) coatings in 1 wt% NaCl: Part II. numerical simulations," *Materials and Corrosion*, vol. 70, pp. 780–792, jan 2019.

- [76] A. S. Vagbharathi and S. Gopalakrishnan, "An extended finite-element model coupled with level set method for analysis of growth of corrosion pits in metallic structures," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 470, p. 20140001, aug 2014.
- [77] R. Duddu, "Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method," *Computational Mechanics*, vol. 54, pp. 613–627, apr 2014.
- [78] Z. Xu, H. Huang, X. Li, and P. Meakin, "Phase field and level set methods for modeling solute precipitation and/or dissolution," *Computer Physics Communications*, vol. 183, pp. 15–19, jan 2012.
- [79] W. Chen and E. D. Schutter, "Parallel STEPS: Large scale stochastic spatial reaction-diffusion simulation with high performance computers," *Frontiers in Neuroinformatics*, vol. 11, feb 2017.
- [80] S. N. Arjunan, A. Miyauchi, K. Iwamoto, and K. Takahashi, "pSpatioocyte: a high-performance simulator for intracellular reaction-diffusion systems," *BMC Bioinformatics*, vol. 21, jan 2020.
- [81] M. J. Hallock, J. E. Stone, E. Roberts, C. Fry, and Z. Luthey-Schulten, "Simulation of reaction diffusion processes over biologically relevant size and time scales using multi-GPU workstations," *Parallel Computing*, vol. 40, pp. 86–99, may 2014.
- [82] B. E. Kendall, C. J. Briggs, W. W. Murdoch, P. Turchin, S. P. Ellner, E. McCauley, R. M. Nisbet, and S. N. Wood, "WHY DO POPULATIONS CYCLE? a SYNTHESIS OF STATISTICAL AND MECHANISTIC MODELING APPROACHES," *Ecology*, vol. 80, pp. 1789–1805, sep 1999.
- [83] J. Ahrens, B. Geveci, and C. Law, "Paraview: An end-user tool for large data visualization," *The visualization handbook*, vol. 717, 2005.
- [84] E. Wendland and H. Schulz, "Numerical experiments on mass lumping for the advection-diffusion equation," *Pesquisa e Tecnologia Minerva*, vol. 2, pp. 227–233, 01 2005.
- [85] M. Badri, P. Jolivet, B. Rousseau, and Y. Favennec, "High performance computation of radiative transfer equation using the finite element method," *Journal of Computational Physics*, vol. 360, pp. 74–92, may 2018.

- [86] H. A. Daas, L. Grigori, P. Jolivet, and P.-H. Tournier, “A multilevel schwarz preconditioner based on a hierarchy of robust coarse spaces,” 2019.
- [87] G. Karypis and V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs,” *SIAM J. Sci. Comput.*, vol. 20, p. 359392, Dec. 1998.
- [88] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015.
- [89] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent, “A fully asynchronous multifrontal solver using distributed dynamic scheduling,” *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 1, pp. 15–41, 2001.
- [90] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, jan 2003.
- [91] I. C. F. Ipsen and C. D. Meyer, “The idea behind krylov methods,” *The American Mathematical Monthly*, vol. 105, pp. 889–899, dec 1998.
- [92] M. R. Hestenes, E. Stiefel, *et al.*, “Methods of conjugate gradients for solving linear systems,” *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
- [93] G. J. Habetler and E. L. Wachspress, “Symmetric successive over-relaxation in solving diffusion difference equations,” *Mathematics of Computation*, pp. 356–362, 1961.
- [94] S. F. McCormick, *Multigrid methods*. SIAM, 1987.
- [95] G. Russo and P. Smereka, “A remark on computing distance functions,” *Journal of Computational Physics*, vol. 163, pp. 51–67, sep 2000.
- [96] N. I. Z. Abidin, B. Rolfe, H. Owen, J. Malisano, D. Martin, J. Hofstetter, P. J. Uggowitzer, and A. Atrens, “The in vivo and in vitro corrosion of high-purity magnesium and magnesium alloys WZ21 and AZ91,” *Corrosion Science*, vol. 75, pp. 354–366, oct 2013.
- [97] J. L. Gustafson, “Reevaluating amdahl's law,” *Communications of the ACM*, vol. 31, pp. 532–533, may 1988.

- [98] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, spring joint computer conference on - AFIPS' 67 (Spring)*, ACM Press, 1967.
- [99] A. Ghai, C. Lu, and X. Jiao, "A comparison of preconditioned krylov subspace methods for large-scale nonsymmetric linear systems," *Numerical Linear Algebra with Applications*, vol. 26, p. e2215, oct 2018.
- [100] A. M. Hassan and M. El-Shenawee, "Parallel implementation of the diffusion-drift algorithm for modeling the electrophysiological activity of breast tumors," *Journal of Parallel and Distributed Computing*, vol. 71, pp. 1011–1023, jul 2011.
- [101] C. Rettinger, C. Godenschwager, S. Eibl, T. Preclik, T. Schruff, R. Frings, and U. Rüde, "Fully resolved simulations of dune formation in riverbeds," in *High Performance Computing* (J. M. Kunkel, R. Yokota, P. Balaji, and D. Keyes, eds.), (Cham), pp. 3–21, Springer International Publishing, 2017.
- [102] M. Barzegari, D. Mei, S. V. Lamaka, and L. Geris, "Computational modeling of degradation process of biodegradable magnesium biomaterials," *Corrosion Science*, vol. 190, p. 109674, 2021.
- [103] M. Barzegari and L. Geris, "Highly scalable numerical simulation of coupled reactiondiffusion systems with moving interfaces," *The International Journal of High Performance Computing Applications*, vol. 36, no. 2, pp. 198–213, 2022.
- [104] D. Höche, "Simulation of corrosion product deposit layer growth on bare magnesium galvanically coupled to aluminum," *Journal of The Electrochemical Society*, vol. 162, no. 1, pp. C1–C11, 2014.
- [105] J. Sanz-Herrera, E. Reina-Romo, and A. Boccaccini, "In silico design of magnesium implants: Macroscopic modeling," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 79, pp. 181–188, mar 2018.
- [106] C. Köhn, D. van Laethem, J. Deconinck, and A. Hubin, "A simulation study of steric effects on the anodic dissolution at high current densities," *Materials and Corrosion*, vol. 72, no. 4, pp. 610–619, 2021.
- [107] T. G. Kurtz, "The relationship between stochastic and deterministic models for chemical reactions," *The Journal of Chemical Physics*, vol. 57, no. 7, pp. 2976–2978, 1972.

- [108] C. Dapogny, C. Dobrzynski, and P. Frey, "Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems," *Journal of Computational Physics*, 2014.
- [109] M. Barzegari and L. Geris, "Biodeg corrosion and biodegradation simulation software: user manual and theory guide," 2022.
- [110] M. Dehghan, "An inverse problem of finding a source parameter in a semilinear parabolic equation," *Applied Mathematical Modelling*, vol. 25, no. 9, pp. 743–754, 2001.
- [111] A. Tarantola, *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier Science Ltd, 1987.
- [112] P. Magni and G. Sparacino, "5 - parameter estimation," in *Modelling Methodology for Physiology and Medicine (Second Edition)* (E. Carson and C. Cobelli, eds.), pp. 83 – 110, Elsevier, second edition ed., 2014.
- [113] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning - Volume 28*, p. I115I123, JMLR.org, 2013.
- [114] O. Chen, S. Kalyuga, and J. Sweller, "The worked example effect, the generation effect, and element interactivity," *Journal of Educational Psychology*, vol. 107, no. 3, p. 689704, 2015.

# **This is curriculum**

...



# List of publications

Input file chapters/publications/publications.tex does not exist. Make sure its starts with “\chapter{List of publications}”. To not include this chapter in the table of contents, use the starred version of the \chapter command...





FACULTY OF ENGINEERING SCIENCE  
DEPARTMENT OF MECHANICAL ENGINEERING  
BIOMECHANICS SECTION  
Celestijnenlaan 300 box 2419  
B-3001 Leuven

