BioDeg

Biodegradation and Corrosion Simulation using Finite Element

Theory Guide

Version 0.8

(generated February 21, 2022)

Mojtaba Barzegari
Liesbet Geris

BioDeg website

KU LEUVEN

Contents

1 Introduction

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, implants and scaffolds in corrosion experiments. It can handle any geometry of desire and supports parallel computing to simulate large scale models.

This document is an overview on the underlying equations, theories, and numerical implementations used in BioDeg. For more information, please refer to the papers published on BioDeg, which can be found in section "Publications and referencing" in the BioDeg repository. Since BioDeg is developed initially for modeling of magnesium (Mg) degradation, this theory guide is focused on Mg.

2 Mathematical models

The biodegradation process can be considered as a reaction-diffusion system [1], 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 [2]. 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 [3]. 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.

2.1 Underlying chemistry

The chemistry of biodegradation of Mg depends considerably on the surrounding solution and the presence of certain ions [2]. In NaCl solutions, the anodic and cathodic reactions as well as the formation and elimination of side corrosion products can be considered as follows [4]:

$$\mathrm{Mg} + 2\mathrm{H_2O} \xrightarrow{k_1} \mathrm{Mg^{2+}} + \mathrm{H_2} + 2\mathrm{OH^-} \xrightarrow{k_1} \mathrm{Mg(OH)_2} + \mathrm{H_2} \tag{1}$$

$$\mathrm{Mg(OH)_2} + 2\mathrm{Cl^-} \xrightarrow{k_2} \mathrm{Mg^{2+}} + 2\mathrm{Cl^-} + 2\mathrm{OH^-}. \tag{2}$$

Reaction 2 is not fully correct from the chemical point of view. In fact, Mg surface is always covered by MgO layer, and $\mathrm{Mg(OH)_2}$ forms on top of that either at atmospheric conditions or during the immersion. The integrity of this MgO layer is undermined by $\mathrm{Cl^-}$ ions, leading to an increase in degradation rate:

$$\mathrm{MgO} + \mathrm{Cl^-} + \mathrm{H_2O} \xrightarrow{k_2} \mathrm{Mg^{2+}} + \mathrm{Cl^-} + 2\mathrm{OH^-}. \tag{3}$$

Although $\mathrm{Cl^-}$ formally does not participate in reaction 3, it reflects the dependence of Mg corrosion rate on $\mathrm{Cl^-}$ concentration. This effect on the rate of degradation has been widely expressed as the effect of $\mathrm{Cl^-}$ on the $\mathrm{Mg(OH)_2}$ in the literature [4, 5]. 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 $\mathrm{Mg(OH)_2}$ only. Moreover, it has been shown recently that oxygen reduction reaction also takes place during

corrosion of Mg [6, 7, 8]. 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 [2], 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. 1.

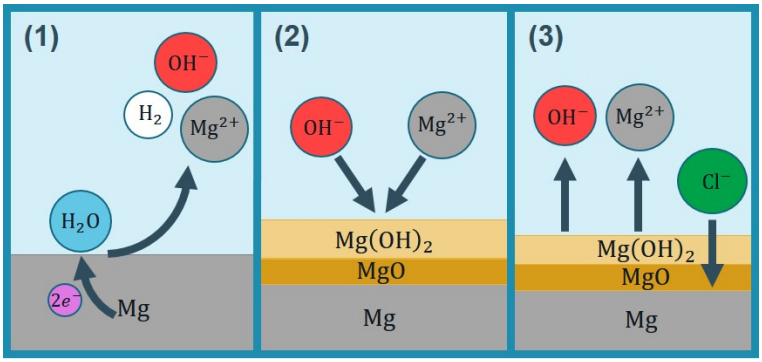

Figure 1: The chemistry of biodegradation of Mg considered in the current study: 1) Mg oxidation and water reduction processes accompanied by releasing Mg^{2+} and OH^- ions as well as H_2 gas, 2) formation of a partially protective precipitation layer, 3) dynamic solubility equilibrium and contribution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

2.2 Mathematical modeling

To keep track of the concentration changes of various contributing chemical components, we define four state variables for the concentration of Mg^{2+} ions, protective film ($Mg(OH)_2$), chloride (Cl^-) ions, and the hydroxide (OH^-) ions:

$$\begin{aligned} &C_{\mathrm{Mg}} = C_{\mathrm{Mg}}(\mathbf{x},t), \quad C_{\mathrm{Film}} = C_{\mathrm{Film}}(\mathbf{x},t) \\ &C_{\mathrm{Cl}} = C_{\mathrm{Cl}}(\mathbf{x},t), \quad C_{\mathrm{OH}} = C_{\mathrm{OH}}(\mathbf{x},t) \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3, \end{aligned} \tag{4}$$

which are indeed 4 scalar functions of space and time. Ω 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 Ω can be calculated as:

$$\mathrm{pH} = 14 + \log_{10} C_{\mathrm{OH}}, \tag{5}$$

where C_{OH} implies the activity of OH^-. By having the definition of the state variables in Eq. 4, the biodegradation of Mg described by Eqs. 1 and 2 can be represented as a set of reaction-diffusion PDEs:

$$\frac{\partial C_{\mathrm{Mg}}}{\partial t} = \nabla \cdot \left(D^e_{\mathrm{Mg}} \nabla C_{\mathrm{Mg}}\right) - k_1 C_{\mathrm{Mg}} \left(1 - \beta \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\max}}\right) + k_2 C_{\mathrm{Film}} {C_{\mathrm{Cl}}}^2 \tag{6}$$

$$\frac{\partial C_{\mathrm{Film}}}{\partial t} = k_1 C_{\mathrm{Mg}} \left(1 - \beta \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\max}}\right) - k_2 C_{\mathrm{Film}} {C_{\mathrm{Cl}}}^2 \tag{7}$$

$$\frac{\partial C_{\mathrm{Cl}}}{\partial t} = \nabla \cdot \left(D^e_{\mathrm{Cl}} \nabla C_{\mathrm{Cl}}\right) \tag{8}$$

$$\frac{\partial C_{\mathrm{OH}}}{\partial t} = \nabla \cdot (D^{e}_{\mathrm{OH}} \nabla C_{\mathrm{OH}}) + k_2 C_{\mathrm{Film}} {C_{\mathrm{Cl}}}^2 \tag{9}$$

in which the maximum concentration of the protective film can be calculated according to its porosity (ϵ) [9]:

$$[\mathrm{Film}]_{\max} = \rho_{\mathrm{Mg(OH)_2}} \times (1 - \epsilon). \tag{10}$$

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 ϵ/τ [10, 11], in which ϵ and τ are the porosity and tortuosity of the protective film, respectively. The effective diffusion coefficient can be then calculated by interpolating the two aforementioned values:

$$D^e_i = D_i \left(\left(1 - \beta \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\max}} \right) + \beta \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\max}} \frac{\epsilon}{\tau} \right). \tag{11}$$

The β coefficient is called momentum here and controls the effect of the saturation term $(1 - \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\max}})$.

2.3 Capturing the moving corrosion interface

In BioDeg, 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 [12]:

$$\frac{\partial \phi}{\partial t} + \overrightarrow{V^{\mathrm{E}}} \cdot \nabla \phi + \mathrm{V^N} |\nabla \phi| = b\kappa |\nabla \phi| \tag{12}$$

in which $\overrightarrow{V^{\mathrm{E}}}$ is the external velocity field, and $\mathrm{V^N}$ 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:

$$\frac{\partial \phi}{\partial t} + \mathrm{V^N} |\nabla \phi| = 0. \tag{13}$$

By having the normal velocity of the interface ($\mathrm{V^N}$) at each point and solving Eq. 13, the interface can be captured at the zero iso-contour of the ϕ function.

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. 13, the interface is obtained at the points with a zero value of the ϕ function. The interface velocity in mass transfer problems can be calculated using the Rankine–Hugoniot equation [13], and by considering the transportation of Mg^{2+} ions, it can be written as:

$$\{\mathbf{J}(x,t) - ([\mathrm{Mg}]_{\mathrm{sol}} - [\mathrm{Mg}]_{\mathrm{sat}})\, \mathrm{V}(x,t)\} \cdot n = 0 \tag{14}$$

where $\mathbf{J}$ is the mass flux at the interface. Rearranging Eq. 14 and inserting the value of the normal interface

velocity into Eq. 13 yields:

$$\frac{\partial \phi}{\partial t} - \frac{D^{e}_{\mathrm{Mg}} \nabla_n C_{\mathrm{Mg}}}{[\mathrm{Mg}]_{\mathrm{sol}} - [\mathrm{Mg}]_{\mathrm{sat}}} |\nabla \phi| = 0, \tag{15}$$

which is the final form of the level set equation to be solved.

3 Computational models

The developed mathematical model comprised of Eqs. 6, 7, 8, 9, and 15 cannot be solved using analytical techniques. The alternative approach in these scenarios is solving the derived PDEs numerically. In BioDeg, 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. 6 is described in detail, but the other PDEs were solved using the same principle.

3.1 Finite element formulation

In order to solve Eq. 6 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_{Mg} is represented as u (the main unknown state variable to find), C_{Film} is denoted by p, C_{Cl} is denoted by q, and the saturation term $(1 - \frac{F}{F_{\max}})$ is denoted by s. By doing this, Eq. 6 can be written as

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - k_1 s u + k_2 p q^2. \tag{16}$$

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

$$\mathcal{V} = \left\{ v(\mathbf{x}) | \mathbf{x} \in \Omega, v(\mathbf{x}) \in \mathcal{H}^1(\Omega), \text{ and } v(\mathbf{x}) = 0 \text{ on } \Gamma \right\} \tag{17}$$

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

$$\mathcal{S}_t = \left\{ u(\mathbf{x}, t) | \mathbf{x} \in \Omega, t > 0, u(\mathbf{x}, t) \in \mathcal{H}^1(\Omega), \text{ and } \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma \right\}. \tag{18}$$

Then, we multiply Eq. 16 to an arbitrary function $v \in \mathcal{V}$:

$$\frac{\partial u}{\partial t} v = \nabla \cdot (D \nabla u) v - k_1 s u v + k_2 p q^2 v. \tag{19}$$

Integrating over the whole domain yields:

$$\int_\Omega \frac{\partial u}{\partial t} v d\omega = \int_\Omega \nabla \cdot (D\nabla u) v d\omega - \int_\Omega k_1 s u v d\omega + \int_\Omega k_2 p q^2 v d\omega. \tag{20}$$

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

$$\int_\Omega \nabla \cdot (D\nabla u) v d\omega = \int_\Omega \nabla \cdot [v(D\nabla u)] d\omega - \int_\Omega (\nabla v) \cdot (D\nabla u) d\omega \tag{21}$$

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

$$\int_\Omega \nabla \cdot [v(D\nabla u)] d\omega = \int_\Gamma D v \frac{\partial u}{\partial n} d\gamma. \tag{22}$$

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:

$$\frac{\partial u}{\partial t} = \frac{u - u^n}{\Delta t} \tag{23}$$

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. 21, 22, and 23 into Eq. 20 yields:

$$\int_\Omega \frac{u - u^n}{\Delta t} v d\omega = \int_\Gamma D v \frac{\partial u}{\partial n} d\gamma - \int_\Omega D\nabla u \cdot \nabla v d\omega - \int_\Omega k_1 s u v d\omega + \int_\Omega k_2 p q^2 v d\omega. \tag{24}$$

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. 18). By reordering the equation, we get the weak form of Eq. 16:

$$\int_\Omega u v d\omega + \int_\Omega \Delta t D\nabla u \cdot \nabla v d\omega + \int_\Omega \Delta t k_1 s u v d\omega = \int_\Omega u^n v d\omega + \int_\Omega \Delta t k_2 p q^2 v d\omega. \tag{25}$$

So, the problem is to find a function $u(t) \in \mathcal{S}_t$ such that for all $v \in \mathcal{V}$ Eq. 25 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. 25 can be simplified as:

$$(u, v)[1 + \Delta t k_1 s] + \Delta t (D\nabla u, \nabla v) = (u^n, v) + \Delta t \, (f^n, v) \tag{26}$$

and so as

$$(u, v) + \Delta t (D\nabla u, \nabla v) + \Delta t k_1 b(u, v) = (u^n, v) + \Delta t \, (f^n, v) \tag{27}$$

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

$$(u, v) + \alpha \Delta t (D\nabla u, \nabla v) = \alpha \, (u^n, v) + \alpha \Delta t \, (f^n, v) \, . \tag{28}$$

One can approximate the unknown function u in Eq. 28 by $u(x) \approx \sum_{i=0}^{N} c_i \psi_i(x)$, where the ψ_i are

the basis functions used to discretize the function space, and $c_0, \ldots, c_N$ 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 $\mathcal{I}_s = \{0, \ldots, N\}$, 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 ψ_i is non-zero inside the ith element and zero everywhere else. In BioDeg, 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 ith element with the width of h can be written as:

$$\psi_i(x) = \begin{cases} 0 & x < x_{i-1} \\ (x - x_{i-1})/h & x_{i-1} \le x < x_i \\ 1 - (x - x_i)/h & x_i \le x < x_{i+1} \\ 0 & x \ge x_{i+1} \end{cases} . \tag{29}$$

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

$$u = \sum_{j=0}^{N} c_j \psi_j(\boldsymbol{x}), \quad u^n = \sum_{j=0}^{N} c_j^n \psi_j(\boldsymbol{x}) \tag{30}$$

as the definition of the unknown function u and its value in the previous time step u^n. We then insert it into Eq. 28, which yields the following equation for each degree of freedom $i = 0, \ldots, N$, where the test functions are selected as $v = \psi_i$:

$$\sum_{j=0}^{N} (\psi_i, \psi_j)\, c_j + \alpha\Delta t \sum_{j=0}^{N} (\nabla\psi_i, D\nabla\psi_j)\, c_j = \sum_{j=0}^{N} \alpha\, (\psi_i, \psi_j)\, c_j^n + \alpha\Delta t\, (f^n, \psi_i)\,. \tag{31}$$

Eq. 31 is a linear system

$$\sum_j A_{i,j} c_j = b_i \tag{32}$$

with

$$A_{i,j} = (\psi_i, \psi_j) + \alpha\Delta t\, (\nabla\psi_i, D\nabla\psi_j) \tag{33}$$

$$b_i = \sum_{j=0}^{N} \alpha\, (\psi_i, \psi_j)\, c_j^n + \alpha\Delta t\, (f^n, \psi_i) \tag{34}$$

which can be rewritten as

$$(M + \alpha\Delta t K)c = \alpha M c_1 + \alpha\Delta t f. \tag{35}$$

M (which traditionally is called the mass matrix), K (which traditionally is called the stiffness matrix),

f, c, and c_1 are defined as

$$
\begin{aligned}
M &= \{M_{i,j}\}\,, \quad M_{i,j} = (\psi_i, \psi_j)\,, \quad i,j \in \mathcal{I}_s \\
K &= \{K_{i,j}\}\,, \quad K_{i,j} = (\nabla\psi_i, D\nabla\psi_j)\,, \quad i,j \in \mathcal{I}_s \\
f &= \{f_i\}\,, \quad f_i = (f(\boldsymbol{x}, t_n)\,, \psi_i)\,, \quad i \in \mathcal{I}_s \\
c &= \{c_i\}\,, \quad i \in \mathcal{I}_s \\
c_1 &= \{c_i^n\}\,, \quad i \in \mathcal{I}_s
\end{aligned}
\tag{36}
$$

By solving Eq. 32 and substituting the obtained c in Eq. 30, u (C_{Mg} in this example) can be calculated in the current time step. As stated before, the same approach can be applied to Eq. 7 and Eq. 15 to get C_{Film} and ϕ. This procedure is repeated in each time step to compute the values of C_{Mg}, C_{Film}, and ϕ over time.

A common practice to save time for solving Eq. 32 for a constant time step size is to compute the left-hand side matrix (A in Eq. 33) 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 α coefficient, the matrix changes along the time. The α 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. 6 and Eq. 16). In addition to this, the diffusion coefficient is not constant (Eq. 11), making the second term in Eq. 33 non-constant even in the absence of α coefficient. Consequently, the left-hand side matrix of the Eq. 32 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.

3.2 Implementation and parallelization

The model was implemented in FreeFEM [14], which is an open-source PDE solver to facilitate converting the weak formulation (Eq. 25) to a linear system $Ax = b$ (with A from Eq. 33 and b from Eq. 34).

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 [15]. 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 in BioDeg 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 [16]. 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 [17]:

$$M^{-1}Ax = M^{-1}b \tag{37}$$

in which the M^{-1} is the preconditioner. In BioDeg, 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) [18]. Then, by defining a positive number δ, the overlapping decomposition $\left\{\mathcal{T}_i^{\delta}\right\}_{1\leqslant i\leqslant N}$ can be created recursively for each sub-mesh $\{\mathcal{T}_i\}_{1\leqslant i\leqslant N}$ by adding all adjacent elements of $\mathcal{T}_i^{\delta-1}$ to it. Then, the finite element space $\mathcal{V}_h$ (Eq. 17) can be mapped to the local space $\left\{\mathcal{V}_i^{\delta}\right\}_{1\leqslant i\leqslant N}$ by considering the restrictions $\{R_i\}_{1\leqslant i\leqslant N}$ and a local partition of unity $\{D_i\}_{1\leqslant i\leqslant N}$ such that:

$$\sum_{j=1}^{N} R_j^{\top} D_j R_j = I_{n\times n} \tag{38}$$

where I and n denote identity matrix and the global number of unknowns, respectively [19].

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

$$M_{\mathrm{RAS}}^{-1} = \sum_{i=1}^{N} R_i^{\top} D_i A_i^{-1} R_i \tag{39}$$

in which $\{A_i\}_{1\leqslant i\leqslant N}$ is the local operator of the sub-matrices [19]. For this purpose, we took advantage of the HPDDM (high-performance domain decomposition methods) package interface in FreeFEM [20]. The partitioned mesh is shown in Fig. 2. The effect of the construction of these local sub-domains on the sparsity pattern of the global matrix is also depicted in Fig. 3. The global matrix is a sparse matrix according to Eq. 33 and the definition of the basis function ψ.

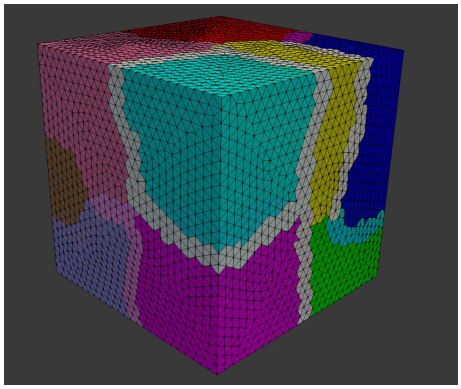

Figure 2: Overlapping domain decomposition in BioDeg. Each color shows a separate sub-domain, and the narrow lighter bands are the overlapped regions.

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 [21]) and iterative solvers (e.g.

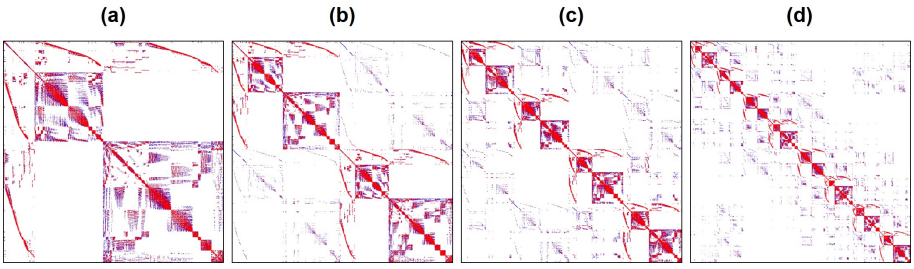

Figure 3: Comparison of the sparsity patterns (highlighting non-zero elements) of the global matrix A for a different number of decomposed domains a: 1 domain b: 2 sub-domains c: 4 sub-domains d: 8 sub-domains.

Generalized Minimal Residual Method, GMRES [22]). 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 [23]. 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 [23]. 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. 37) and then solved it with an iterative solver.

Krylov methods have been frequently used by researchers as robust iterative approaches to parallelism [24]. 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 BioDeg, we used the PETSc (Portable, Extensible Toolkit for Scientific Computation) library [25], 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, and the best performance for the reaction-diffusion system model was achieved using the HYPRE BoomerAMG preconditioner [26] and the GMRES solver [22].

References

[1] 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.

[2] 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.

[3] P. Grindrod, *The theory and applications of reaction-diffusion equations : patterns and waves*. Oxford University Press, 1996.

[4] Y. Zheng, X. Gu, and F. Witte, "Biodegradable metals," *Materials Science and Engineering: R: Reports*, vol. 77, pp. 1–34, mar 2014.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] P. Grathwohl, *Diffusion in Natural Porous Media: Contaminant Transport, Sorption/Desorption and Dissolution Kinetics*. Springer US, 1998.

[11] 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.

[12] S. O. Ronald Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer New York, 2002.

[13] 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.

[14] F. Hecht, "New development in freefem++," *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012.

[15] 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.

[16] 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.

[17] H. A. Daas, L. Grigori, P. Jolivet, and P.-H. Tournier, "A multilevel schwarz preconditioner based on a hierarchy of robust coarse spaces," 2019.

[18] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, p. 359–392, Dec. 1998.

[19] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015.

[20] P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme, "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.

[21] 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.

[22] 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.

[23] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, jan 2003.

[24] I. C. F. Ipsen and C. D. Meyer, "The idea behind krylov methods," *The American Mathematical Monthly*, vol. 105, pp. 889–899, dec 1998.

[25] 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.

[26] 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.