BioDeg: corrosion/biodegradation simulation software for metallic biomaterials based on FreeFEM / PETSc / Qt

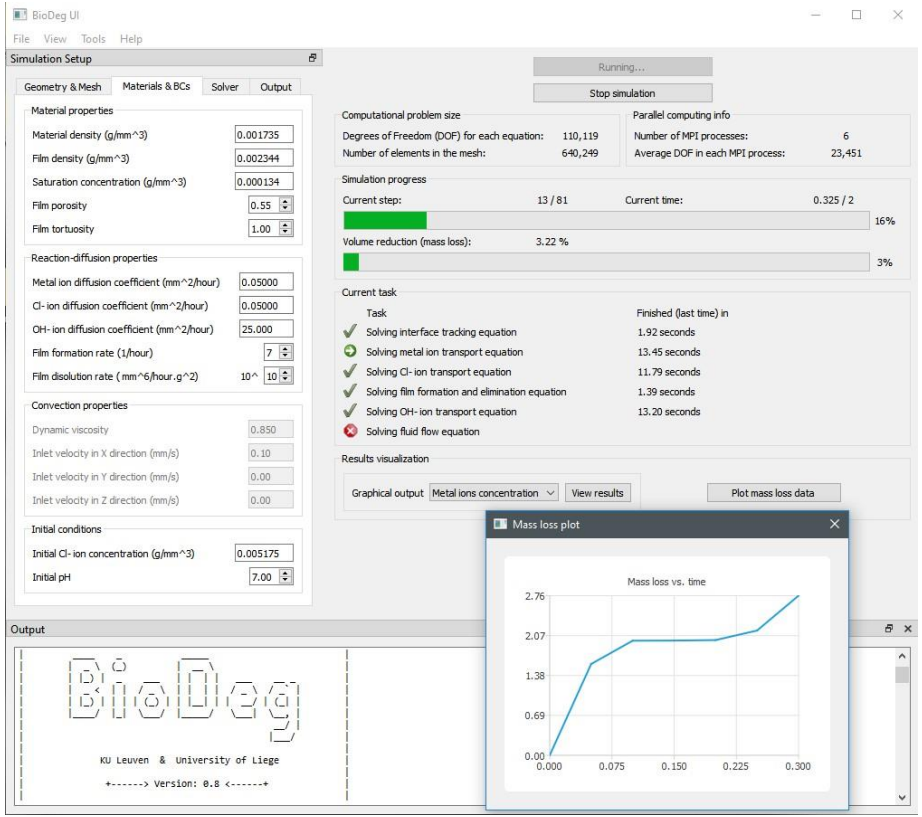
Mojtaba Barzegari, Liesbet Geris

Biomechanics Section, Department of Mechanical Engineering,

KU Leuven, Leuven, Belgium

BioDeg Software

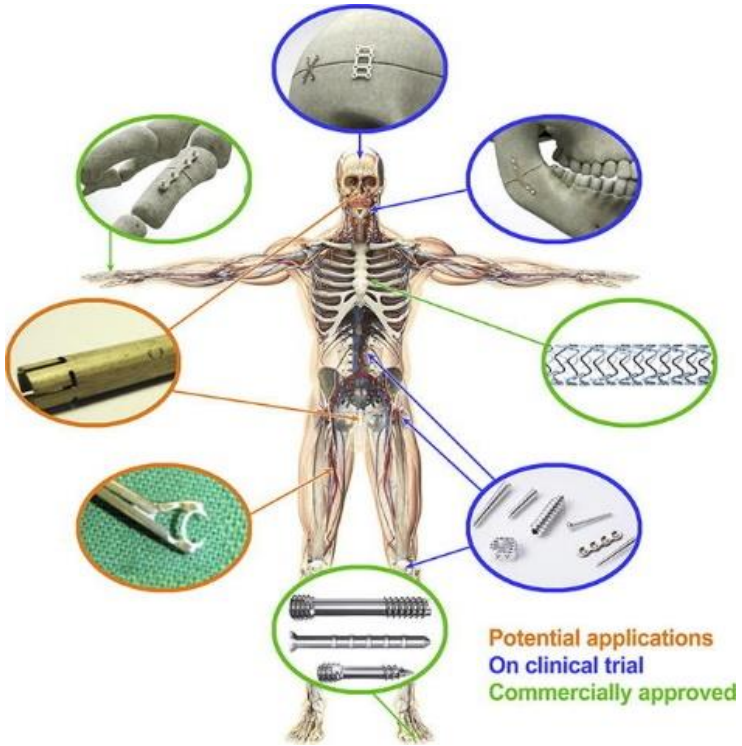
- Multifunctional 3D simulation code for modeling biodegradation
- Cross-platform user interface
- Included pre- and post-processors
- FeeFEM/PETSc backend
  Qt frontend
- Available as an open source software



Statement of Need
& Basic Concepts

Biodegradable Metals

- Mg, Zn, and Fe
- Gradually disappear/absorbed and replaced by new tissue/bone
- Great mechanical/biological properties
- The controlled release profile is an issue for different types of implants
- The degradation behavior should be tuned/optimized for various applications

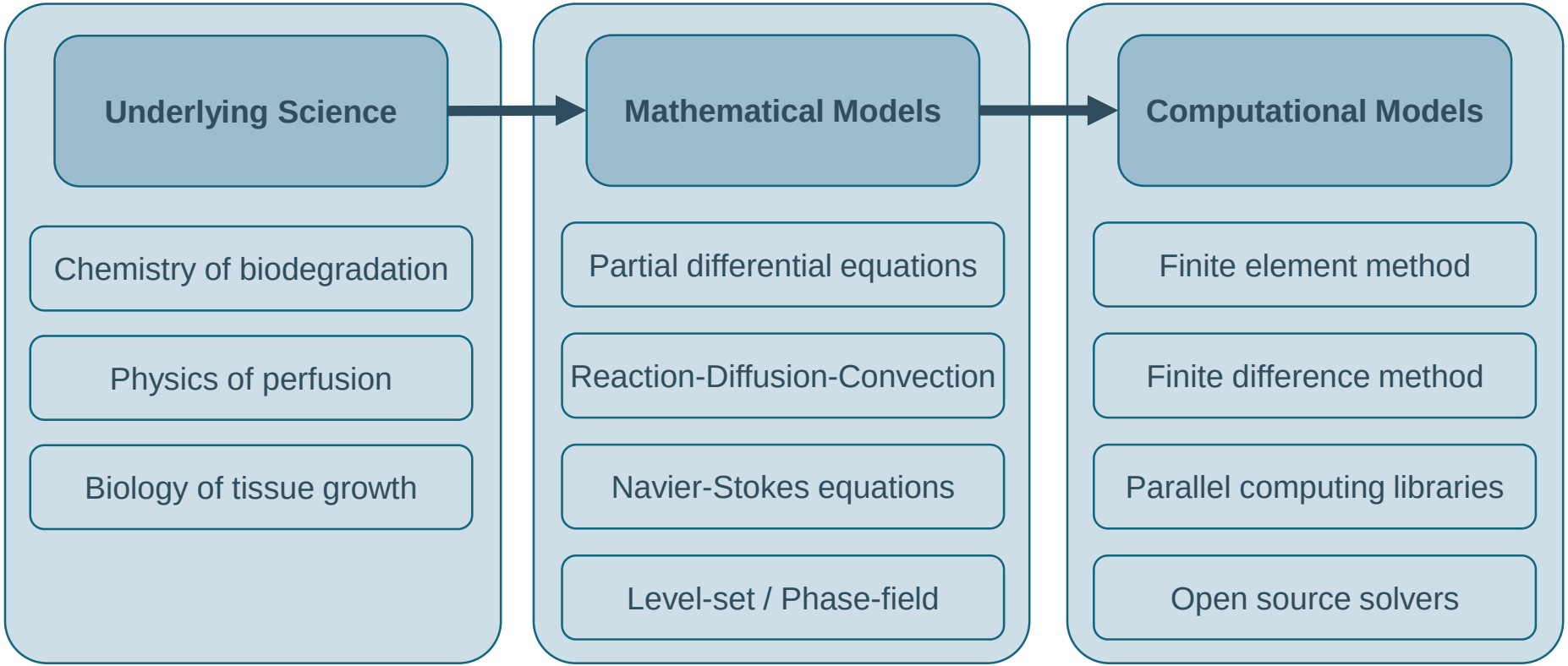


(Han et al., Mater. Today, 23, 2019)

Problem Definition

- Challenge:
  - Tuning the biodegradation to the regeneration of the new tissue/bone
  - Optimizing the implant design based on the release profile
- Can be solved by:
  - Mathematical modeling of biodegradation
  - Coupling biodegradation models with tissue growth models
  - Considering environmental effects

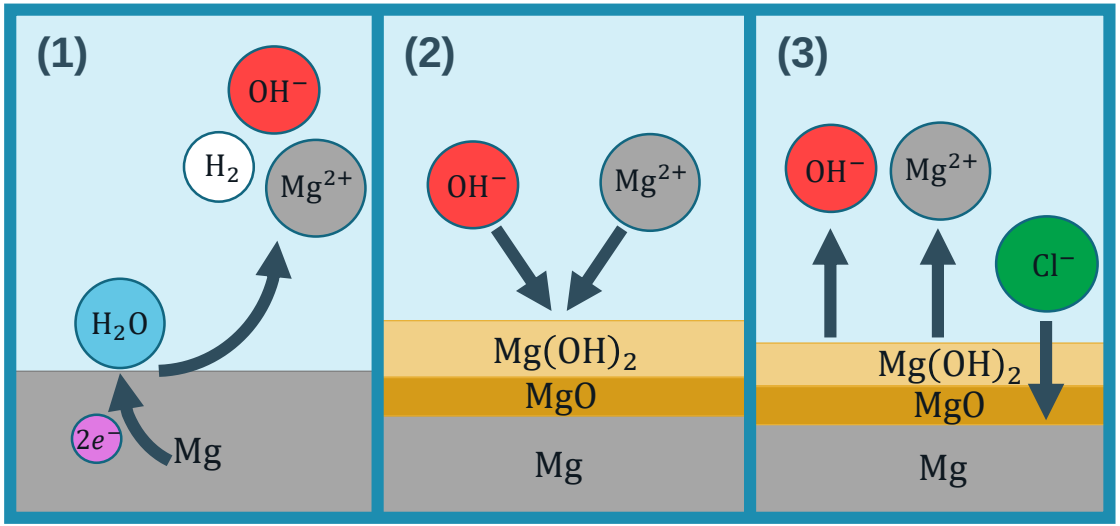
Modeling Workflow



Chemistry of Biodegradation

The model captures:

1. The chemistry of dissolution of metallic implant
2. Formation of a protective film
3. Effect of ions in the medium
4. Change of pH



Mathematical Representation

Chemical reactions

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

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

State variables

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

Derived Partial Differential Equations

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

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

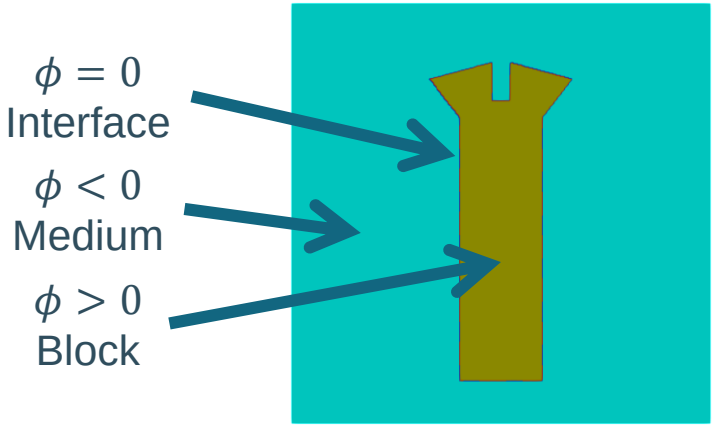
$$\frac{\partial C_{Cl}}{\partial t} = \nabla.\left(D^e_{Cl}\nabla C_{Cl}\right) - \nabla.\left(\mathbf{v}C_{Cl}\right)$$

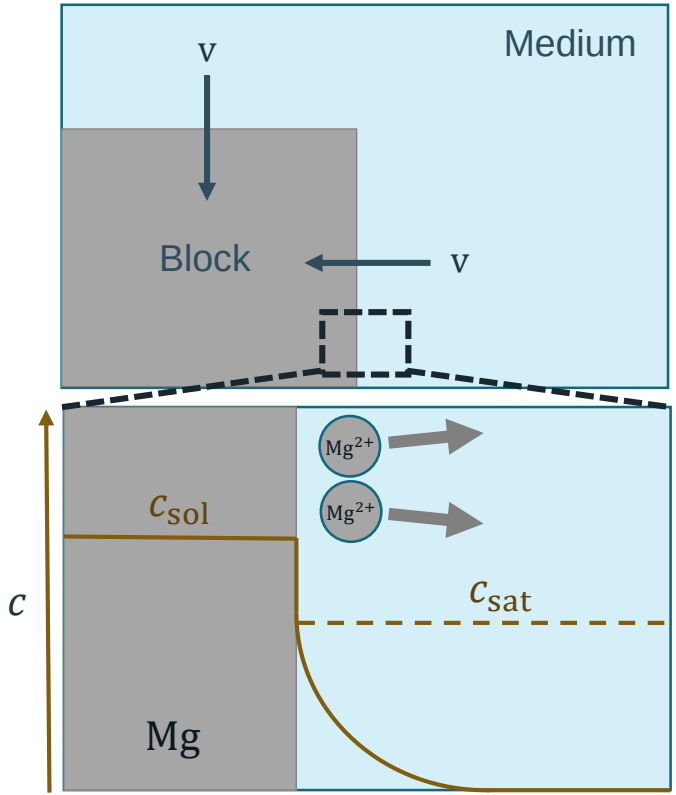
$$\frac{\partial C_{OH}}{\partial t} = \nabla.\left(D^e_{OH}\nabla C_{OH}\right) - \nabla.\left(\mathbf{v}C_{OH}\right) + k_2 C_{Film}{C_{Cl}}^2$$

Capturing the Moving Biodegradation Interface

- Implicit moving interface tracking
- Level set method

PDE to solve: $$\frac{\partial \phi}{\partial t} - \frac{D^e_{\mathrm{Mg}} \nabla_n C_{\mathrm{Mg}}}{[\mathrm{Mg}]_{\mathrm{sol}} - [\mathrm{Mg}]_{\mathrm{sat}}} |\nabla \phi| = 0$$



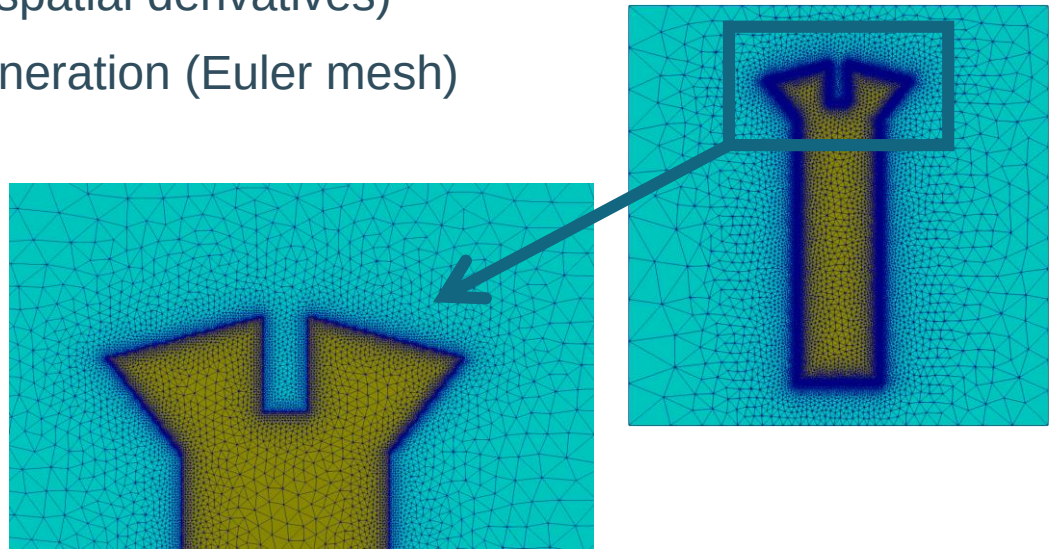


KU LEUVEN

Computational Model Implementation

Constructing Computational Model

- Discretizing PDE equations, numerical computation
  - Finite difference method (temporal derivatives)
  - Finite element method (spatial derivatives)
- Adaptively refined mesh generation (Euler mesh)



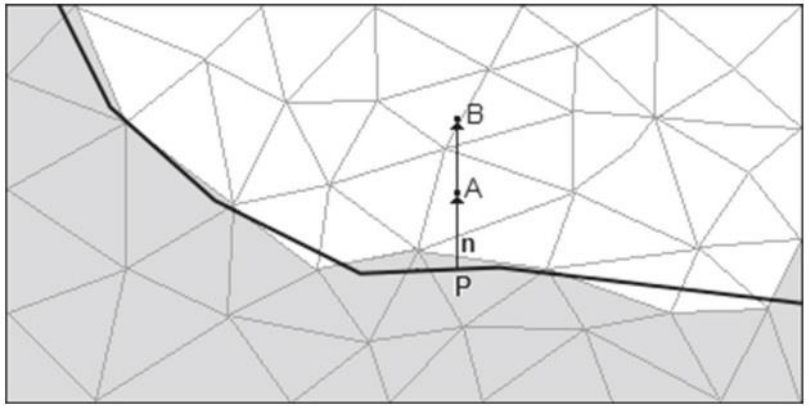
Implementing Computational Model

- Mesh generation (SALOME, Mmg), #Tetrahedra ~ 10M – 20M
- Weak form implementation (FreeFEM), #DoF of each PDE ~ 2M – 4M
- Parallelization is essential
  - High-performance domain decomposition (HPDDM)
  - High-performance preconditioners and solvers (PETSc)
- Paralleled IO postprocessing (ParaView)

Level Set Implementation

- Penalization for interface BCs
- Problem of concentration oscillation
- Computing $\nabla_n C_{\mathrm{Mg}}$ correctly
- Problem of re-distancing the distance function ϕ

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



(Bajger et al., Biomech. Model., 2017)

Considering Convection

- Adding fluid flow and considering the effect of hydrodynamics condition
- Requires dealing with Navier-Stokes equations

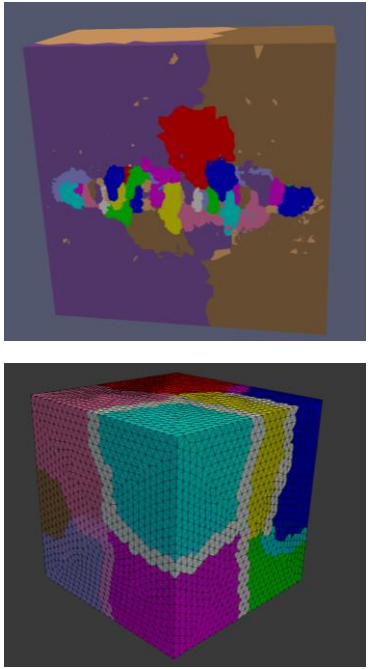
$$\begin{cases} \rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - \mu \nabla^2 \mathbf{u} + \nabla p = 0 \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

- Implementing a parallel fluid flow code in FreeFEM (with fieldsplit preconditioner)
- Comparing the output of the CFD code with OpenFOAM (simpleFOAM)

https://github.com/mbarzegary/navier-stokes-solver

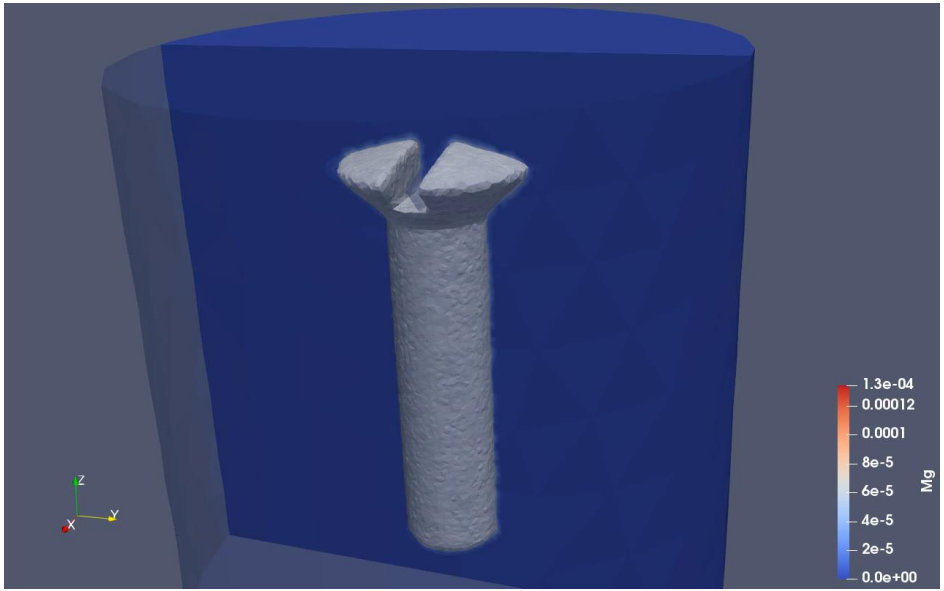
Parallelizing Computation

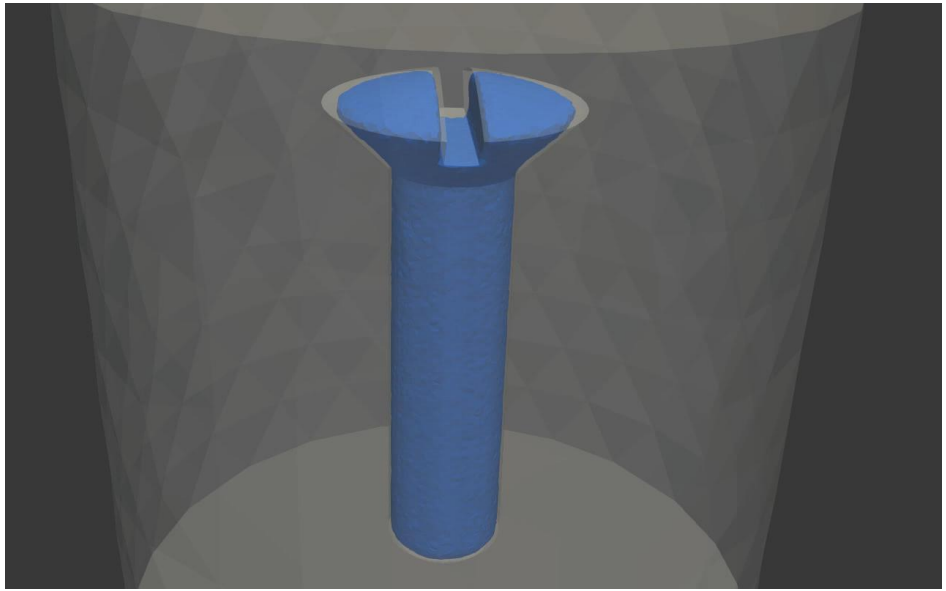
- Distributing the mesh among available resources
  - High-performance mesh decomposition
  - Overlapping Schwarz method
- Solving the linear system of equations
  - HYPRE BoomerAMG preconditioner
  - Fieldsplit preconditioner (for fluid flow)
  - GMRES iterative solver



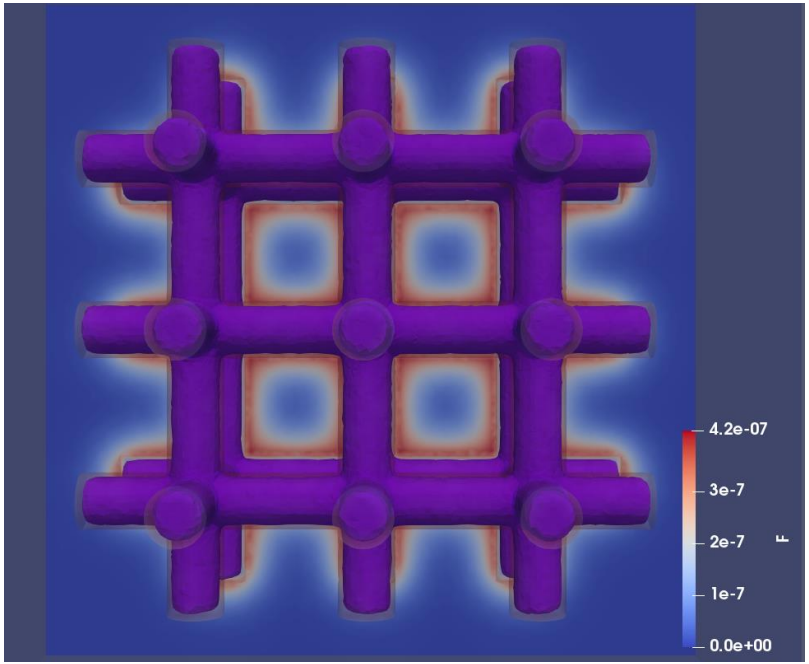
Simulation and Parallelization Results

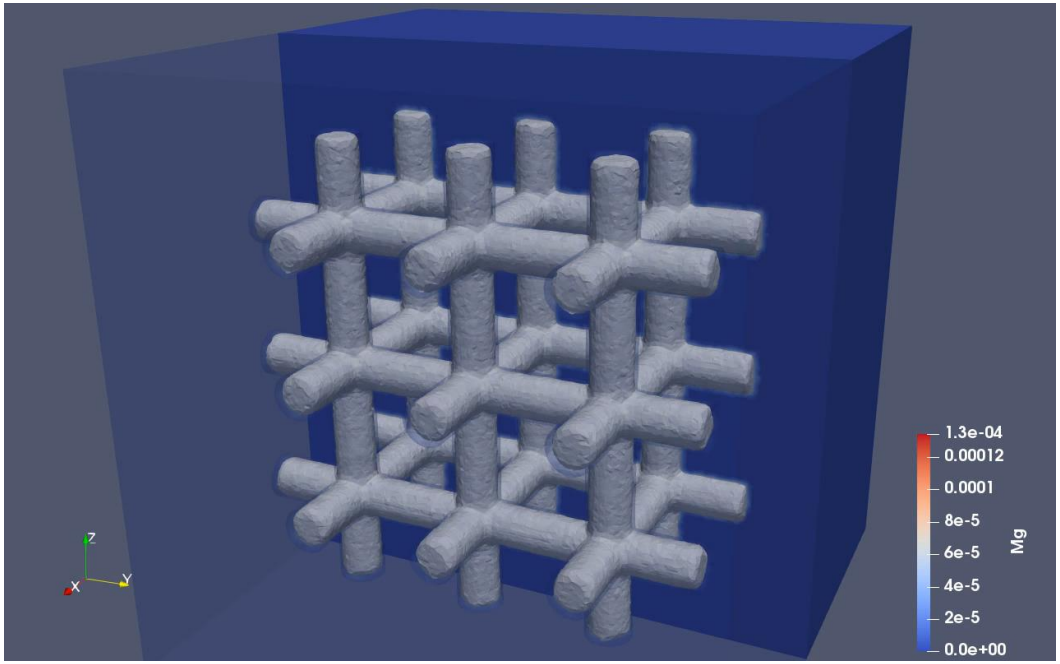
Orthopedics Screw Degradation



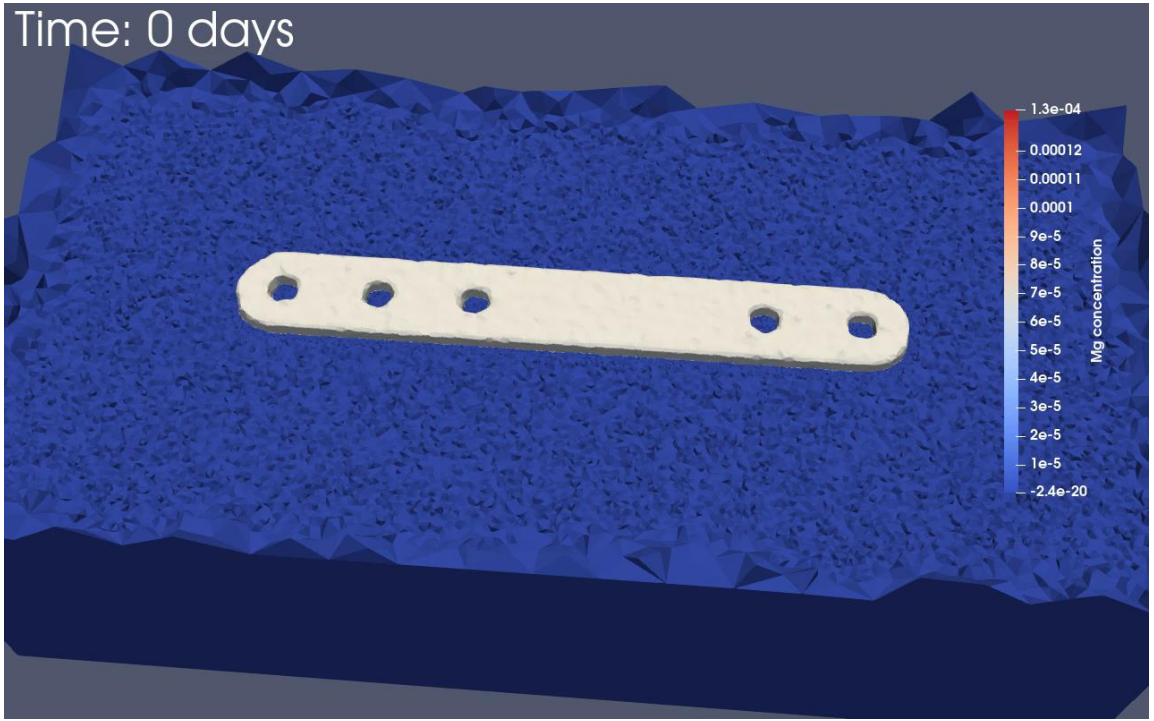


Porous Scaffold Degradation

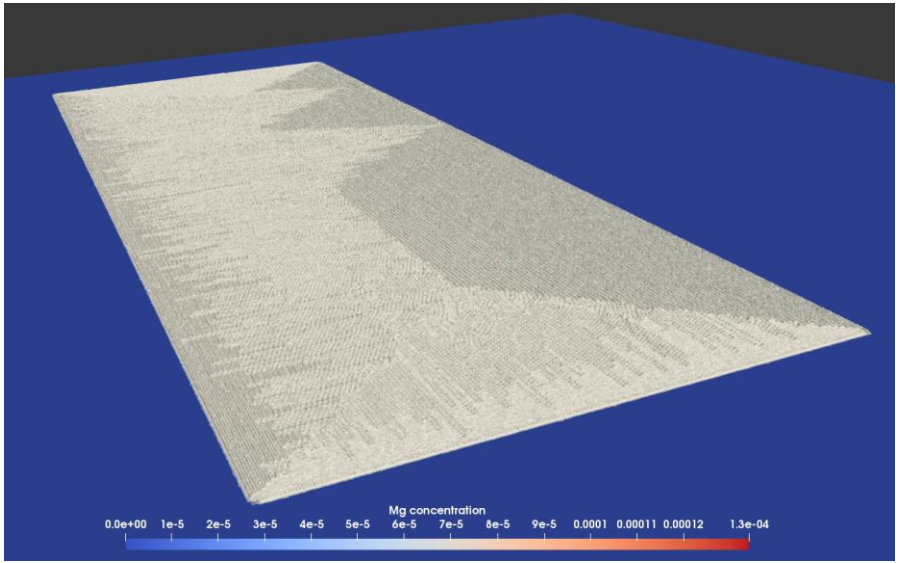


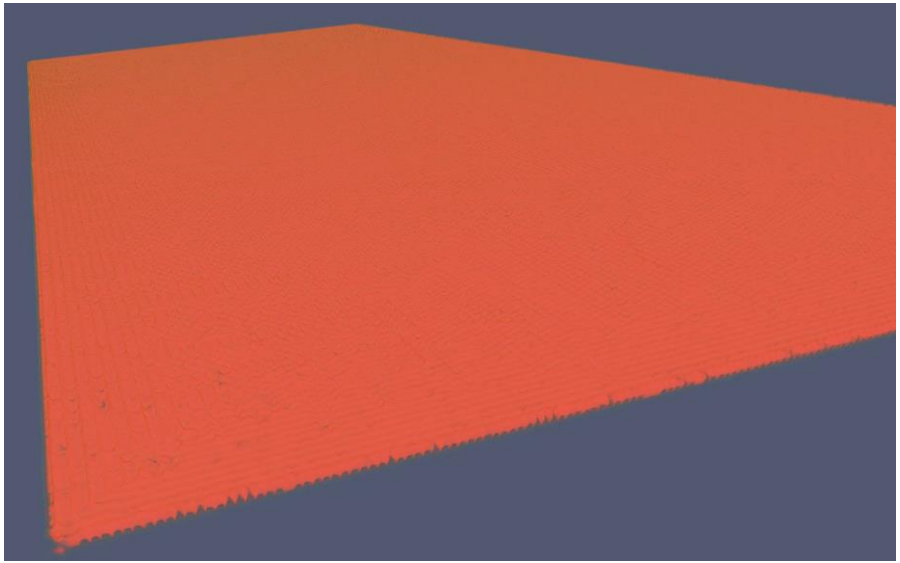


Jaw Bone Plate Degradation



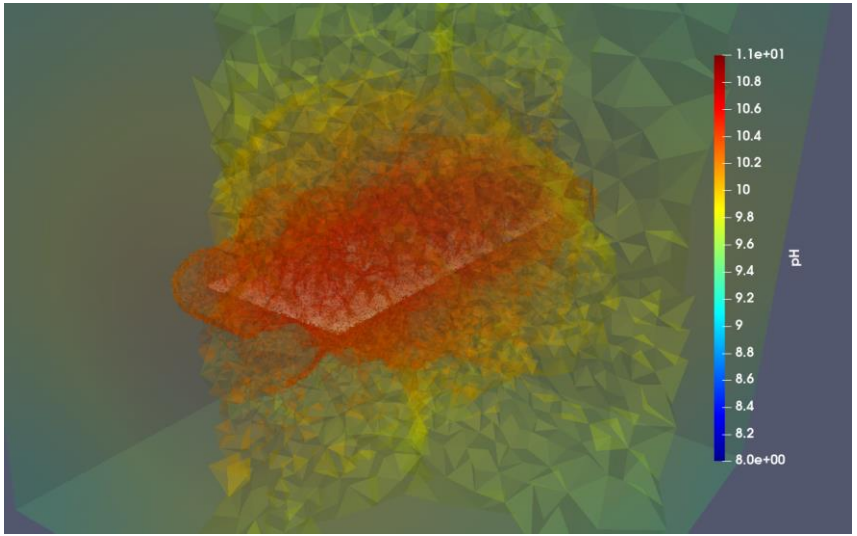
Narrow Cuboid



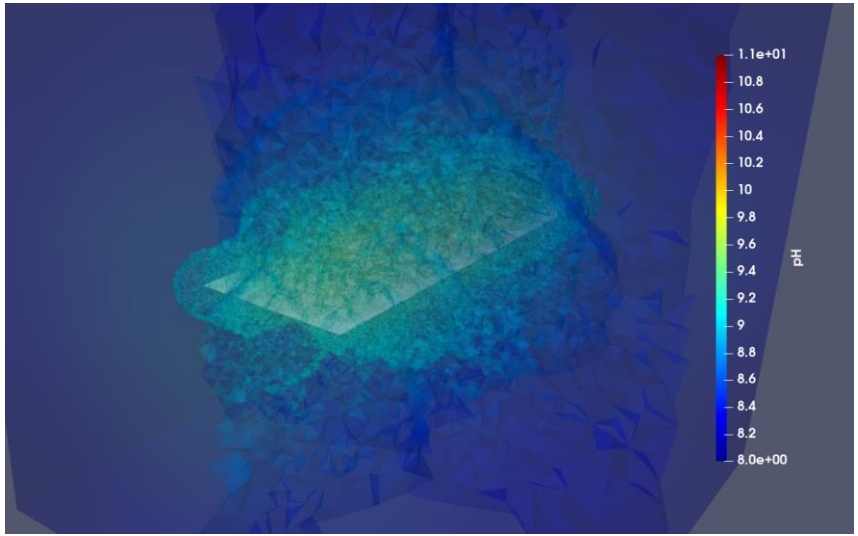


(Barzegari et al., Corrosion Science, 190, 2021)

Narrow Cuboid - pH Change



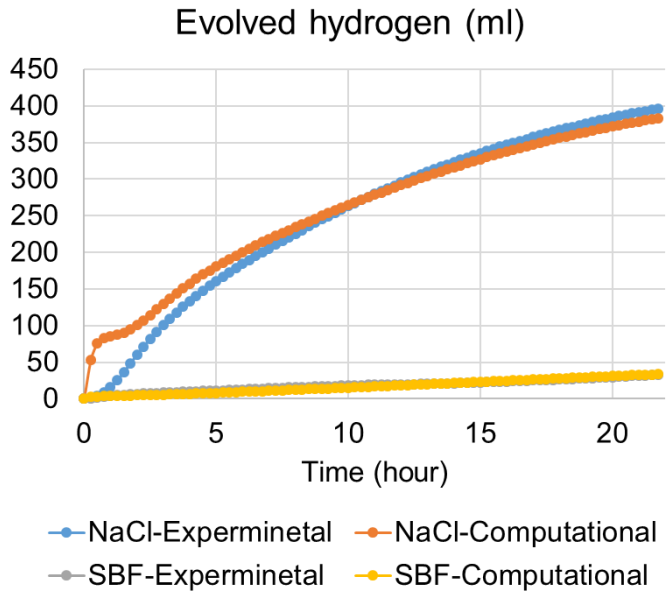
High diffusion (Saline solution)

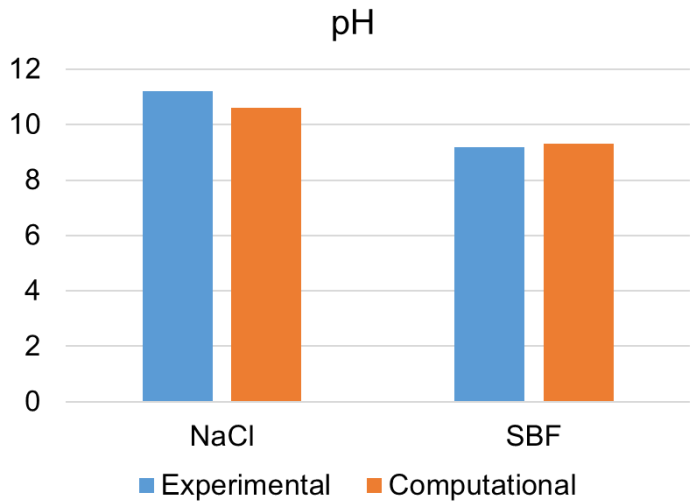


Low diffusion (Buffered solution)

(Barzegari et al., Corrosion Science, 190, 2021)

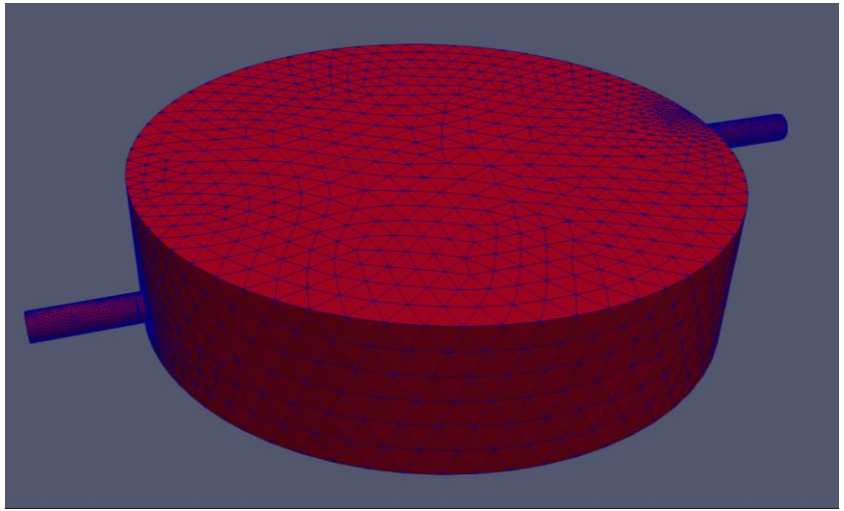
Quantitative Results for Validation



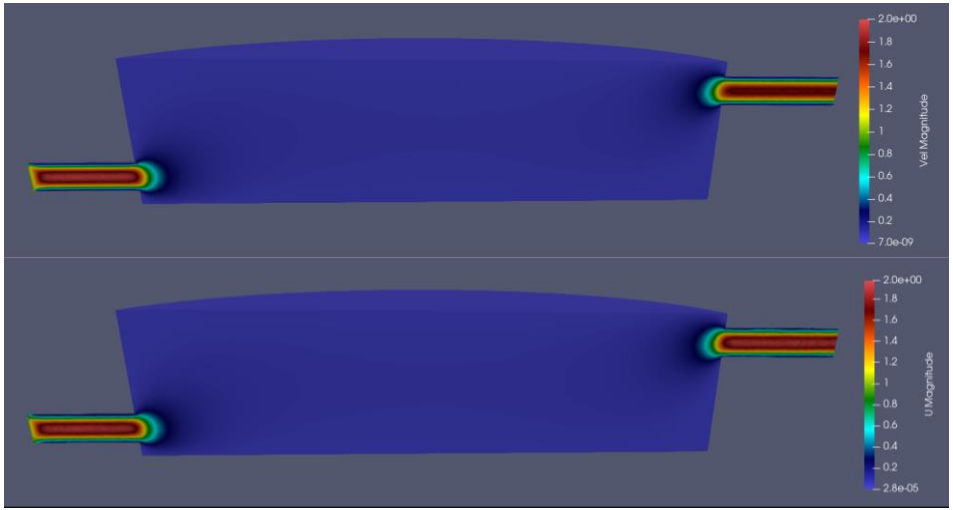


(Barzegari et al., Corrosion Science, 190, 2021)

Comparing Fluid Results with OpenFOAM

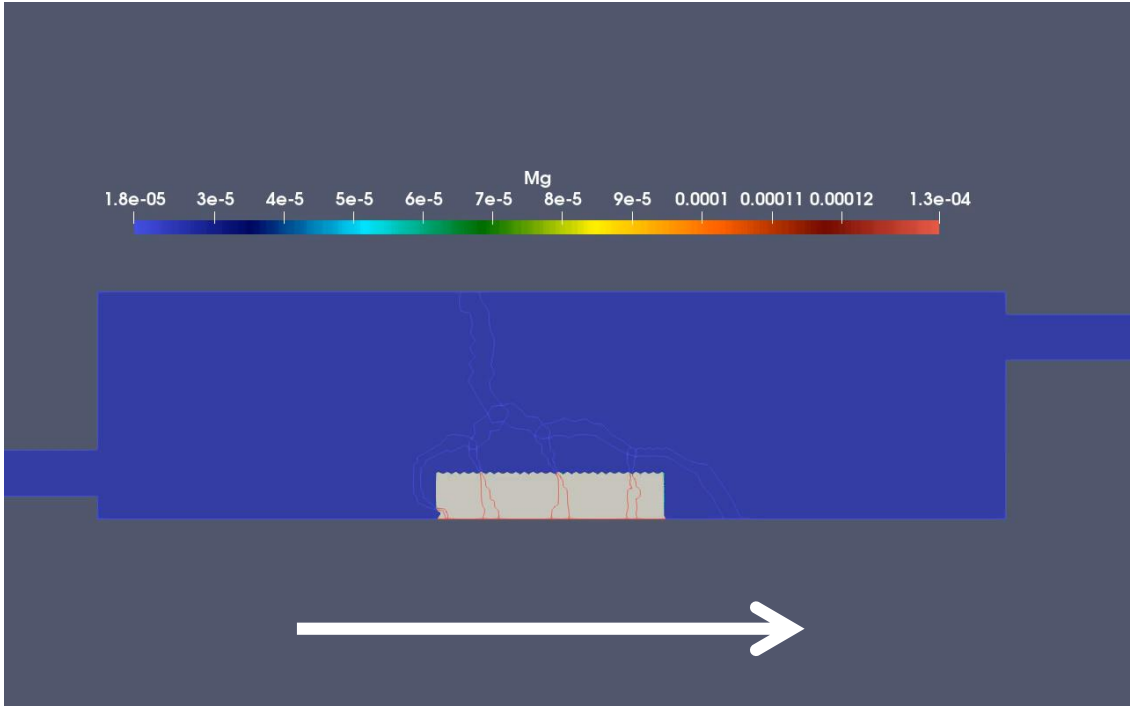


The chamber with inlet and outlet,
used to simulate hydrodynamics condition



Fluid velocity magnitude
(top: in-house code, bottom: OpenFOAM)

Degradation with Flow

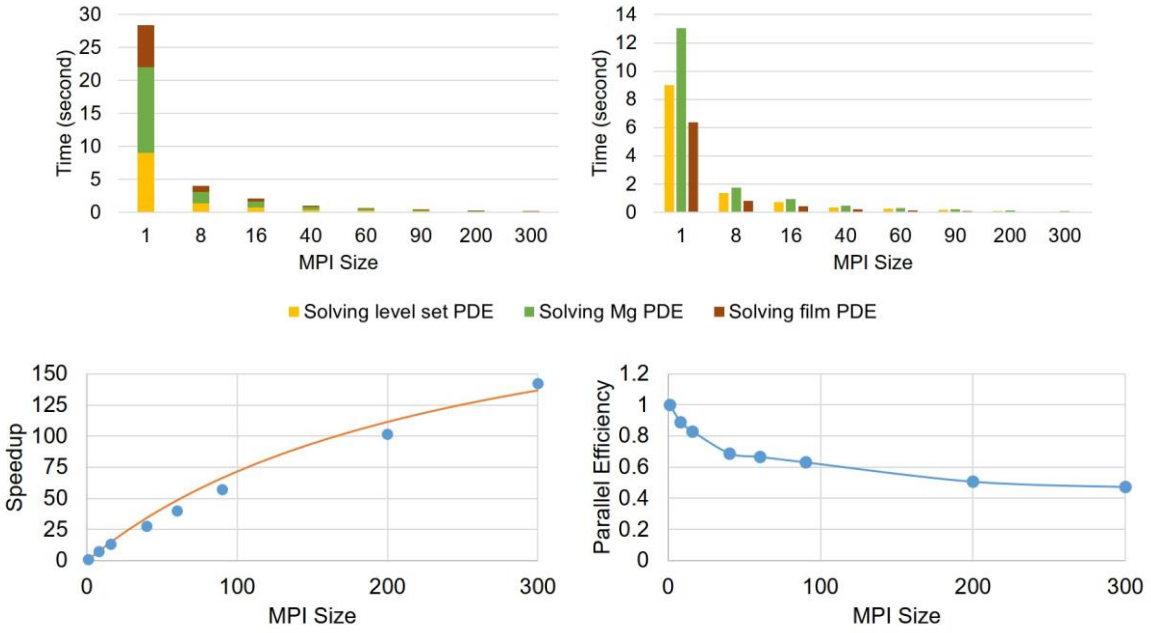


Parallel Strong Scaling Analysis (# of tetrahedra ~ 900K)

Amdahl's law

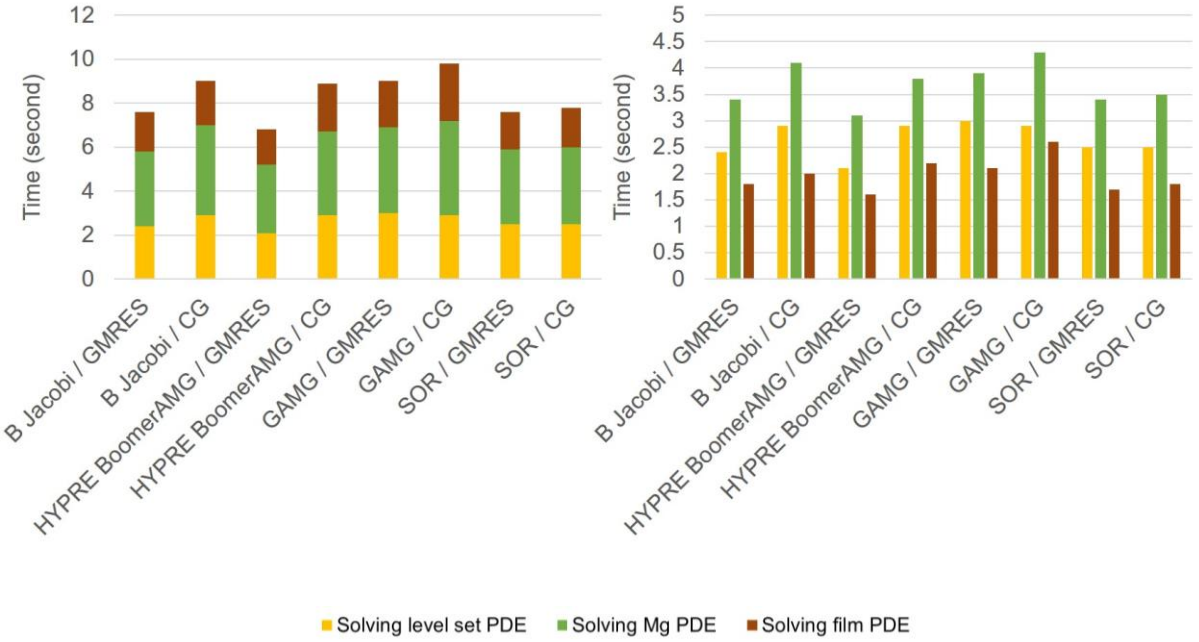
$$\text{Speedup} = \frac{1}{f + \frac{1-f}{N}}$$

- Sequential part: 1%
- Parallel part: 99%



(Barzegari et al., Int. J. High. Perform. Comp., 35, 2021)

Preconditioner/Solver Performance

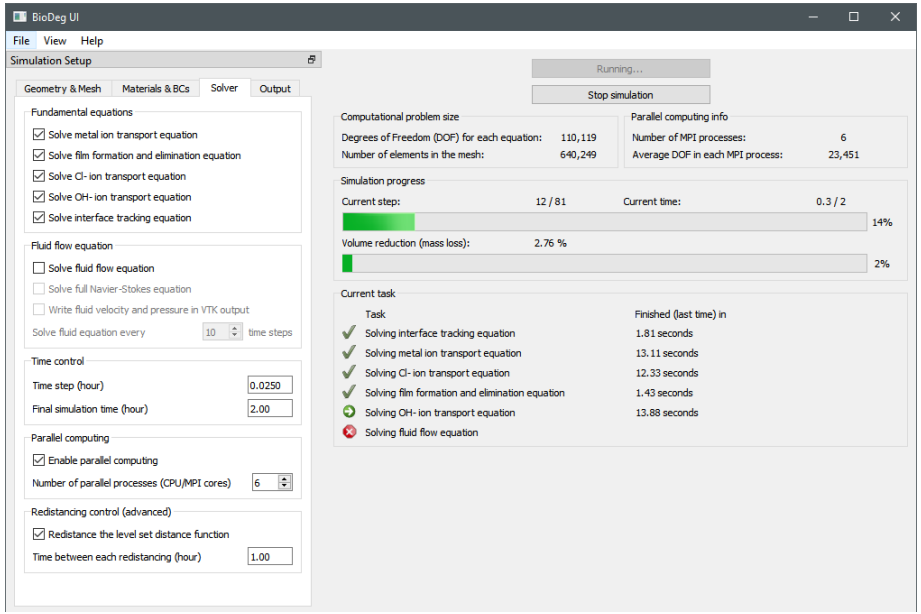


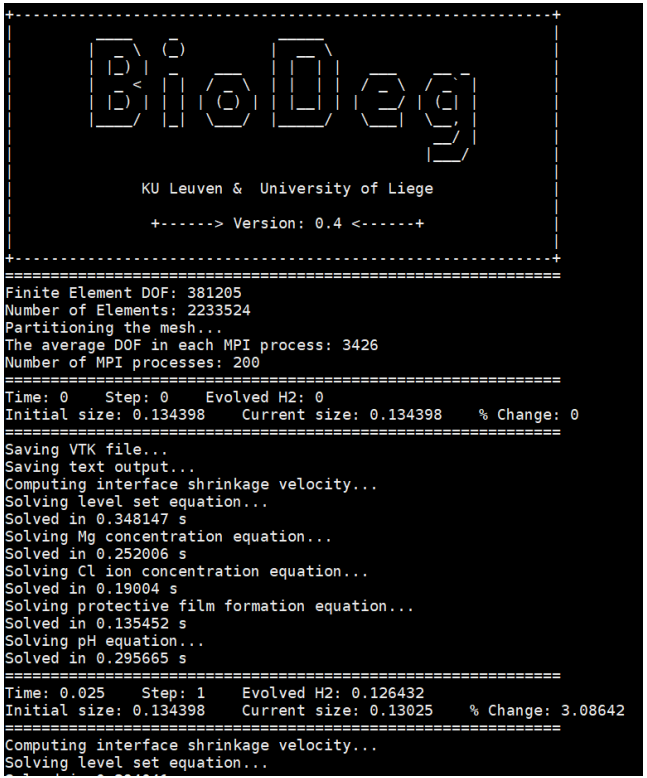
(Barzegari et al., Int. J. High. Perform. Comp., 35, 2021)

Software Development

BioDeg: An Interface on FreeFEM Code

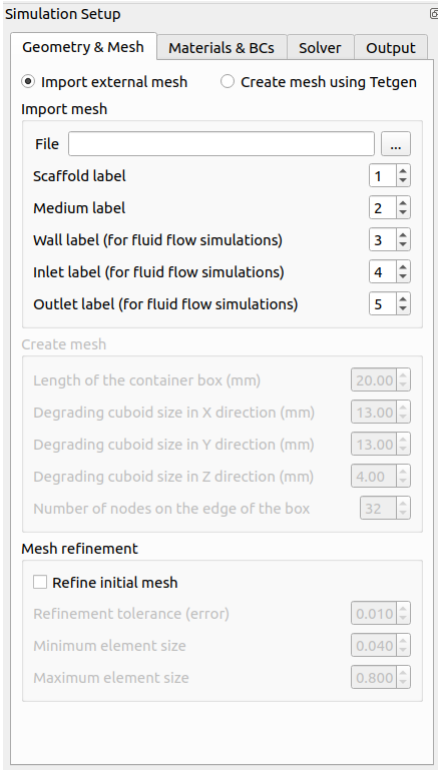
- Cross-platform, customizable, and adaptive UI developed using Qt and C++

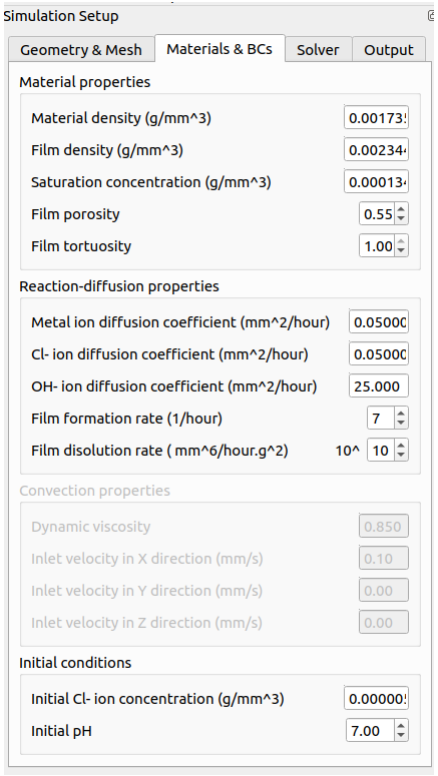


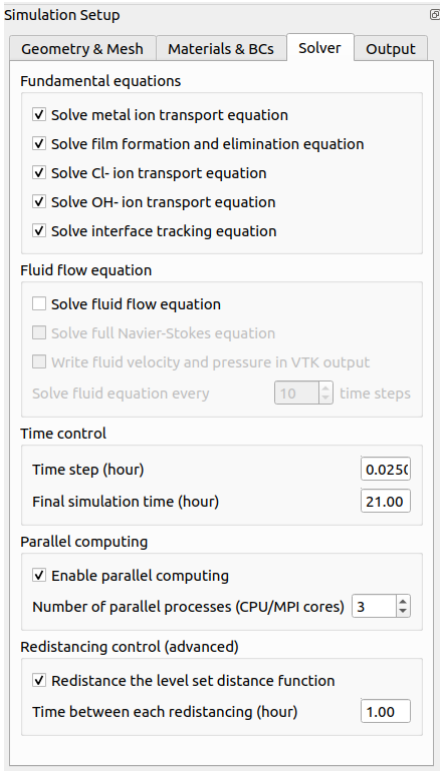


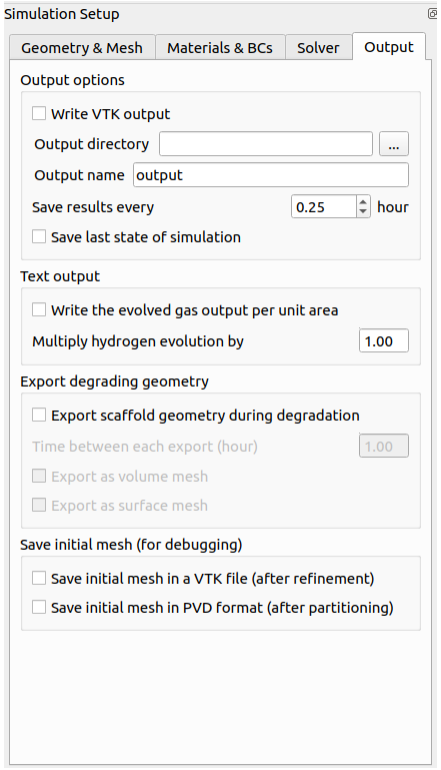
KU LEUVEN

Configuring the Simulation





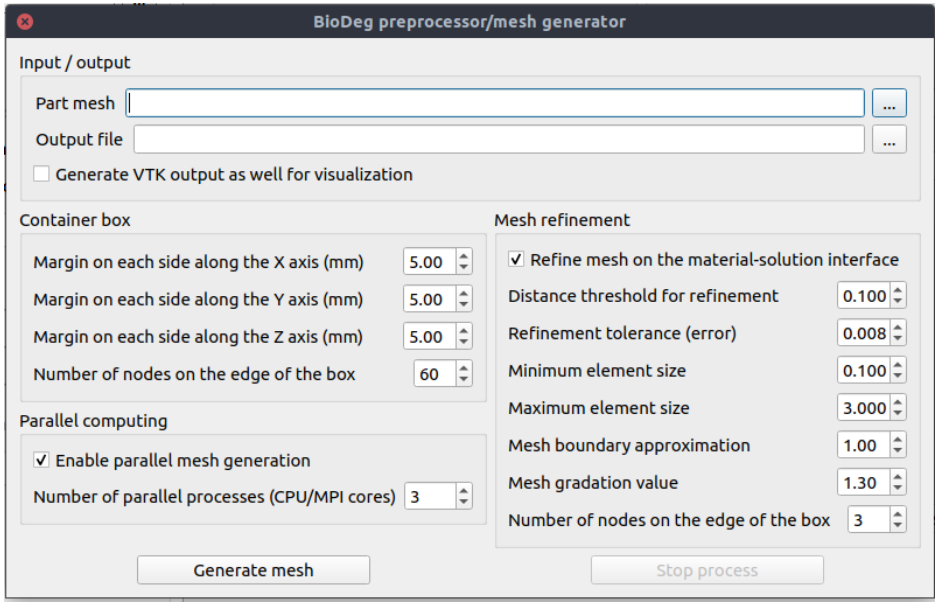




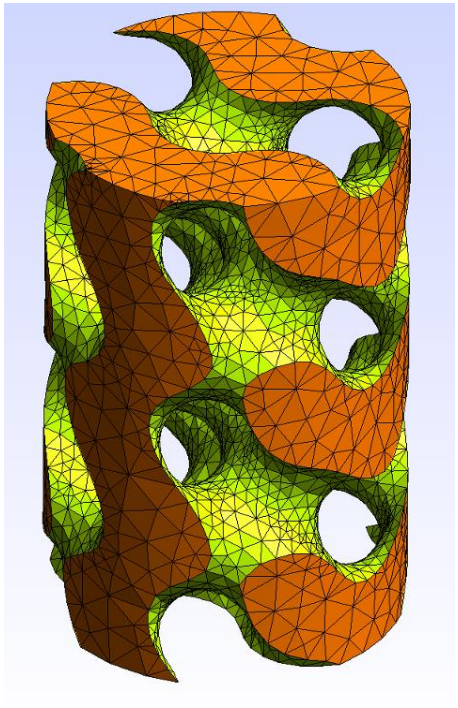
KU LEUVEN

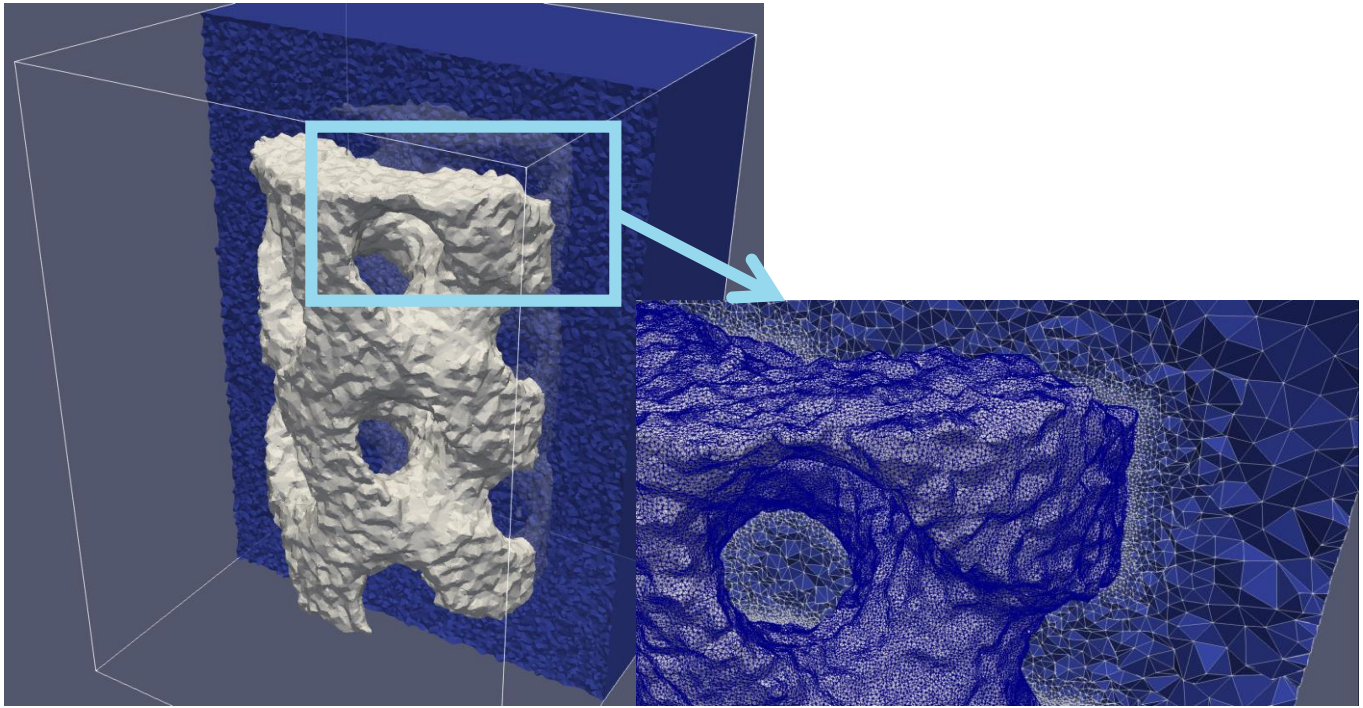
BioDeg Preprocessor

- Developed using ParMmg and FreeFEM to embed a mesh into a container
- Mesh refinement on the corrosion interface



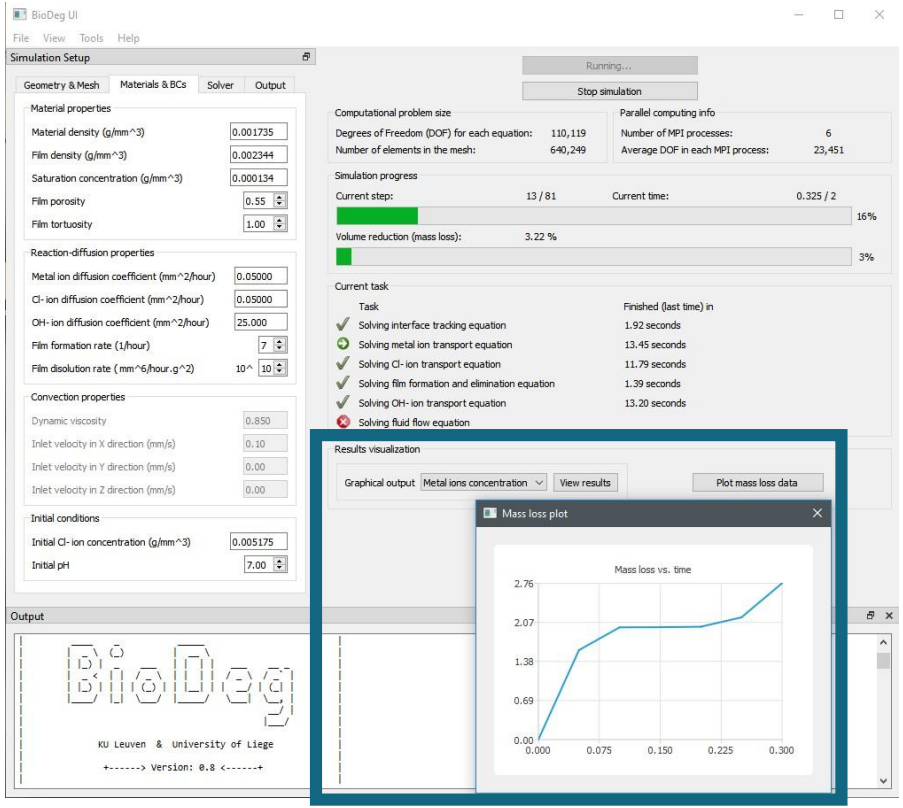
BioDeg Preprocessor





BioDeg Postporcessor

- Developed using Python and Qt Charts
- Basic integration at this moment
- Will consider better integration options like ParaView Glance in the future



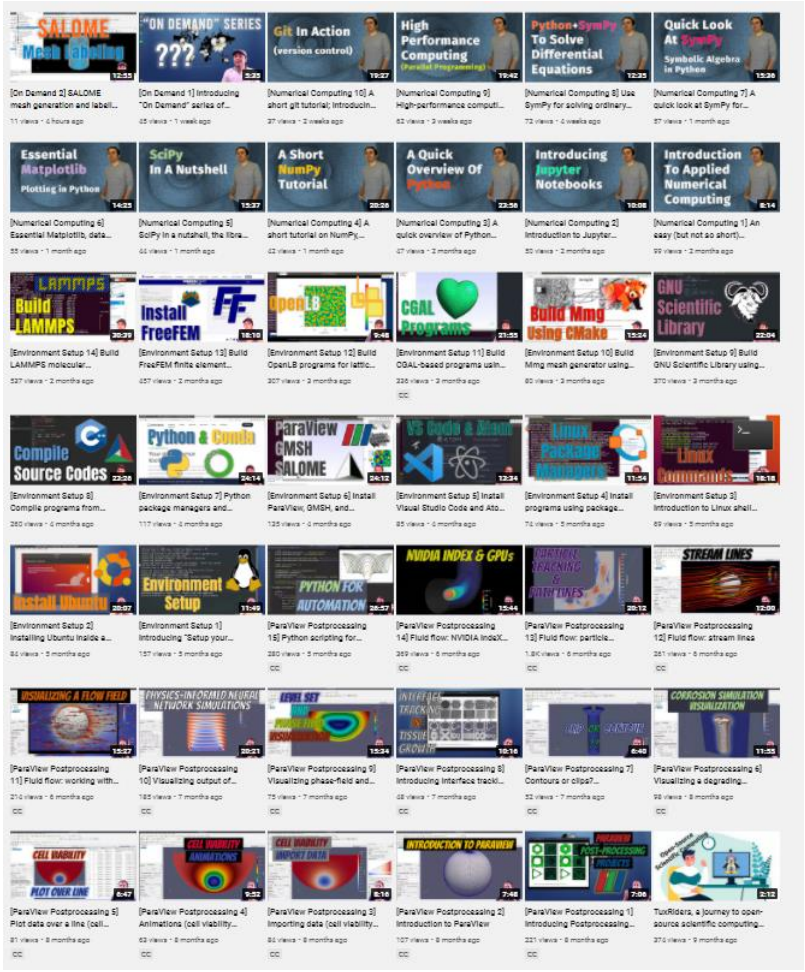
Open Source and Open Science

Hosted on GitHub

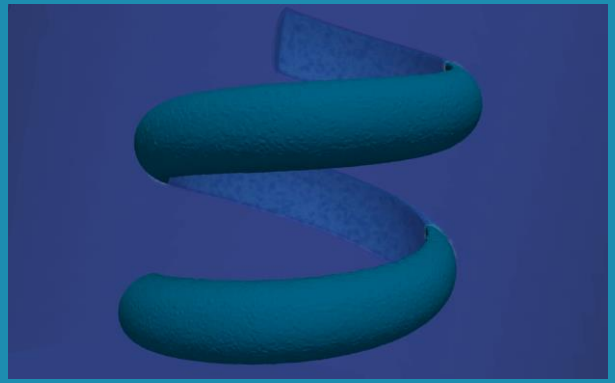
- BioDeg core (FreeFEM, PETSc)  https://github.com/mbarzegary/BioDeg
- BioDeg UI (Qt, C++)  https://github.com/mbarzegary/BioDeg-UI
- BioDeg pre-processor (FreeFEM)  https://github.com/mbarzegary/BioDeg-preprocess
- BioDeg docs  https://github.com/mbarzegary/BioDeg-doc

A Final Note ☺

- A project to share experiences on scientific computing using open-source tools
- More info: TuxRiders.com
  YouTube.com/TuxRiders



KU LEUVEN



Thank you for your attention

https://mbarzegary.github.io

@MojBarz

This research was financially supported by the PROSPEROS project, funded by the Interreg VA Flanders - The Netherlands program

KU LEUVEN