KU LEUVEN

High-Performance Computing in Biomedical Engineering; a Use-case for Biomaterials Degradation Modeling

Mojtaba Barzegari, Liesbet Geris

Biomechanics Section, Department of Mechanical Engineering,

KU Leuven, Leuven, Belgium

High-Performance Computing (HPC)

- Driven by need for more computing power
- Leveraging (distributed) compute resources to solve complex problems
- Computations in parallel over lots of compute elements (software)
- Very fast network to connect compute elements (hardware)
- Why do we need HPC?
  - Complete a time-consuming operation in less time
  - Perform a high number of operations per seconds

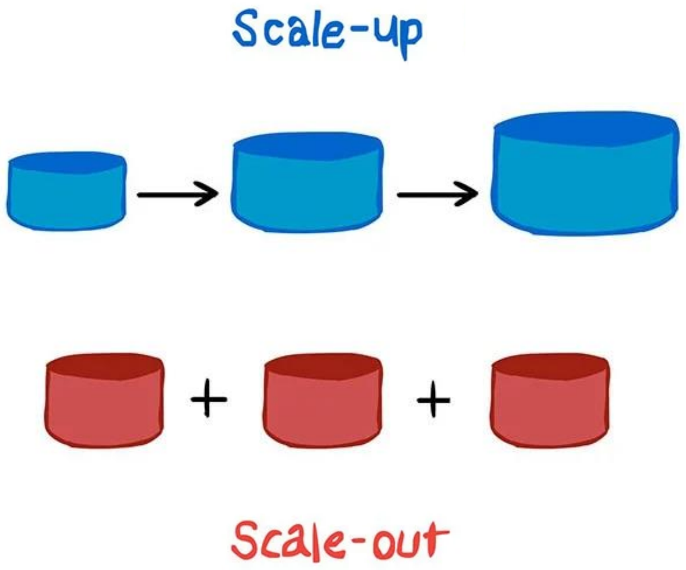
(source: tech.eu)

Typical HPC Workloads

- Physics, astrophysics
- Bioscience, genomics, bioinformatics
- Artificial intelligence
- Chemistry, molecular sciences
- Computer aided engineering
- Computer science, mathematics
- Big data analytics
- Financial and economic modeling
- Weather and climate
- Cyber security
- Advanced graphics
- …

Supercomputing in Computational Sciences

- Solving more complex problems in less time
- Scale up
  - Faster CPUs
  - Larger memories
  - More advanced hardware/software
- Scale out
  - Large parallel applications
  - Many small- to medium- size jobs

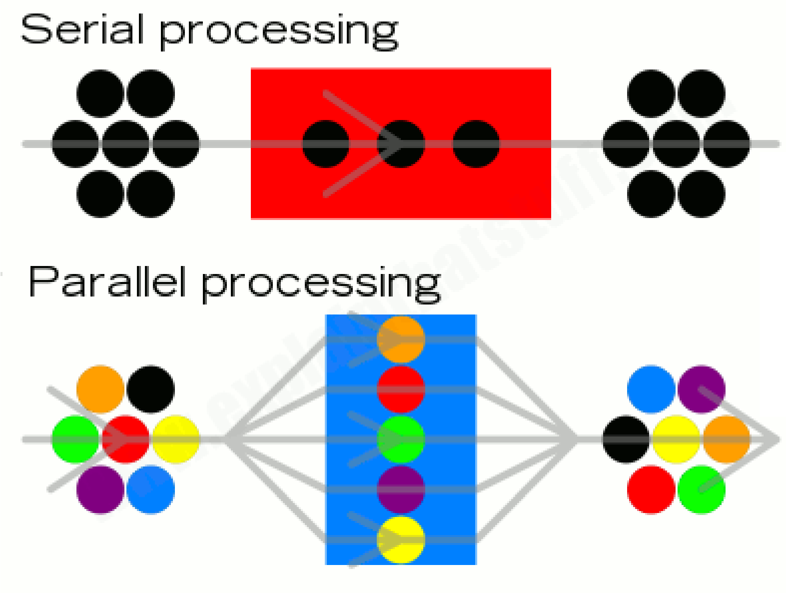


(source: turbonomic.com)

KU LEUVEN

Synonymous to Parallel Computing

- Parallelism is available at many levels
- All HPC machines are parallel architectures
- Two fundamental parallel architectures:
  - Shared memory systems
  - Distributed memory systems
- Scaling benchmark matters a lot!



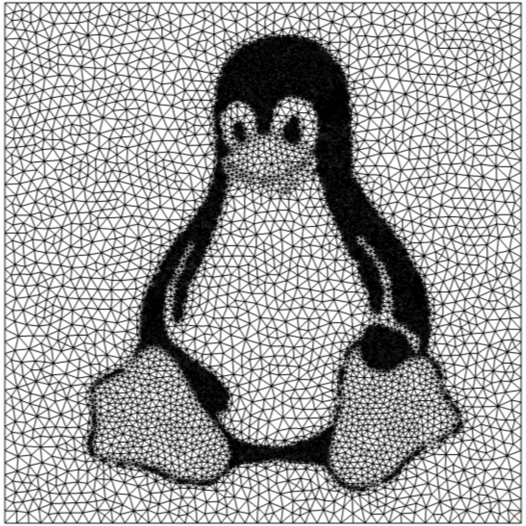
(source: explainthatstuff.com)

HPC in Biomedicine and Biomedical Engineering

- Much effort is put into model credibility and regulatory context of the simulation predictions
- Scalability of models to benefit from rapidly growing computing power
- The potential impact of HPC:
  - employing more quantities of data or parameters
  - using them to perform predictive modelling and simulation
  - in order to deliver therapies and to enhance clinical decision making
  - on time scales far shorter than those in academic research activities

Role of Free and Open Source Software

- The role that open-source movement has played is undeniable
- Freedom and flexibility in HPC
- Huge number of relevant programs and libraries
  - Operating system and system tools
  - Job scheduling and load balancing systems
  - High-performance libraries, tools, and solvers
- Most scientific contributions and codes are open-source



KU LEUVEN

Computational modeling of biodegradation behavior of degradable metals

A high-performance computing project use-case

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
- Can be solved by:
  - Mathematical modeling of biodegradation
  - Coupling biodegradation models with tissue growth models
  - Considering environmental effects

Modeling Workflow

Underlying Science

- Chemistry of biodegradation
- Physics of perfusion
- Biology of tissue growth

→

Mathematical Models

- Partial differential equations
- Reaction-Diffusion-Convection
- Navier-Stokes equations
- Level set method

→

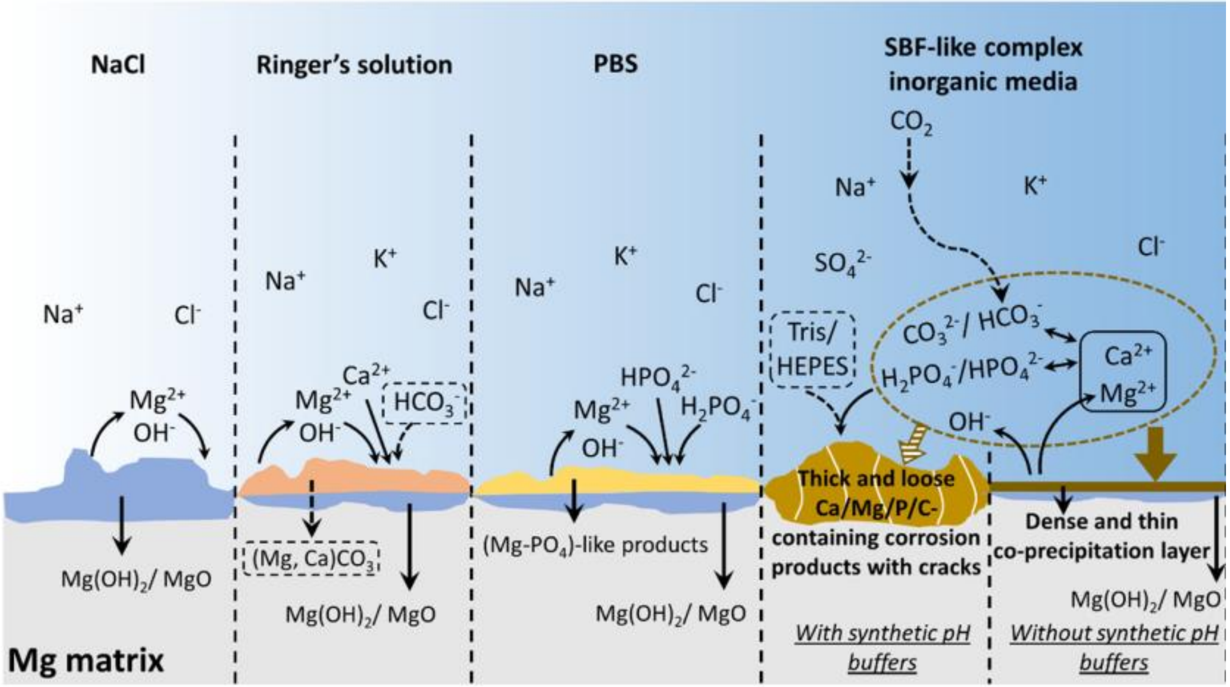
Computational Models

- Finite element method
- Finite difference method
- Scientific computing libraries
- Open source solvers

Chemistry of Biodegradation

The model captures:

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



(Mei et al., Corrosion Science 171, 2020)

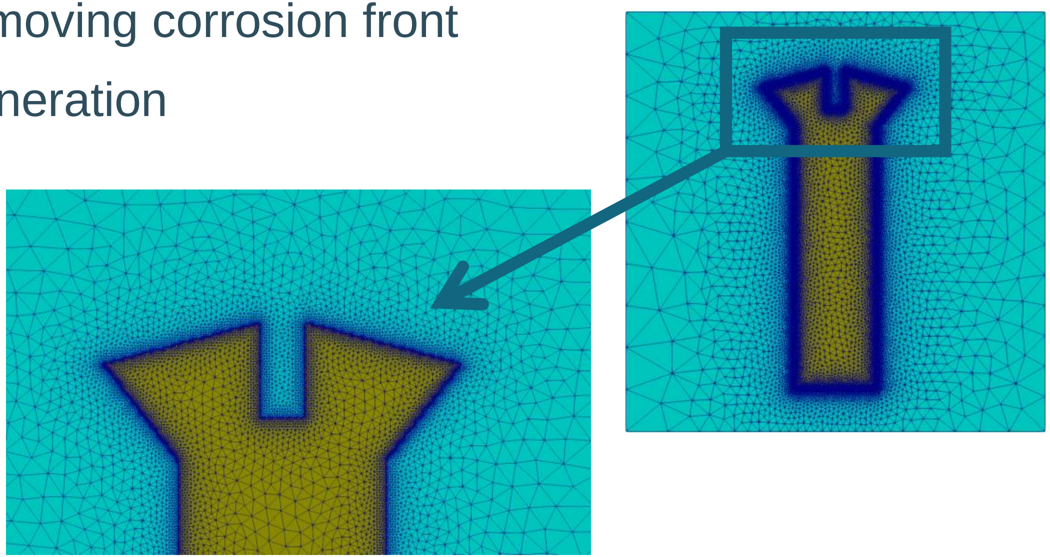
Constructing Mathematical Model

- Converting the chemical interaction into mathematical forms
- Reaction-diffusion-convection partial differential equations (PDE)
- An example for the transport of Mg ions

$$\frac{\partial C_{\mathrm{Mg}}}{\partial t} = \underbrace{\nabla.\left(D_{\mathrm{Mg}}^{e} \nabla C_{\mathrm{Mg}}\right)}_{\text{Diffusion}} - \underbrace{\nabla.\left(\mathrm{v} C_{\mathrm{Mg}}\right)}_{\text{Convection}} - \underbrace{k_1 C_{\mathrm{Mg}} \left(1 - \beta \frac{C_{\mathrm{Film}}}{[\mathrm{Film}]_{\mathrm{max}}}\right) + k_2 C_{\mathrm{Film}} {C_{\mathrm{Cl}}}^{2}}_{\text{Reaction}}$$

Constructing Computational Model

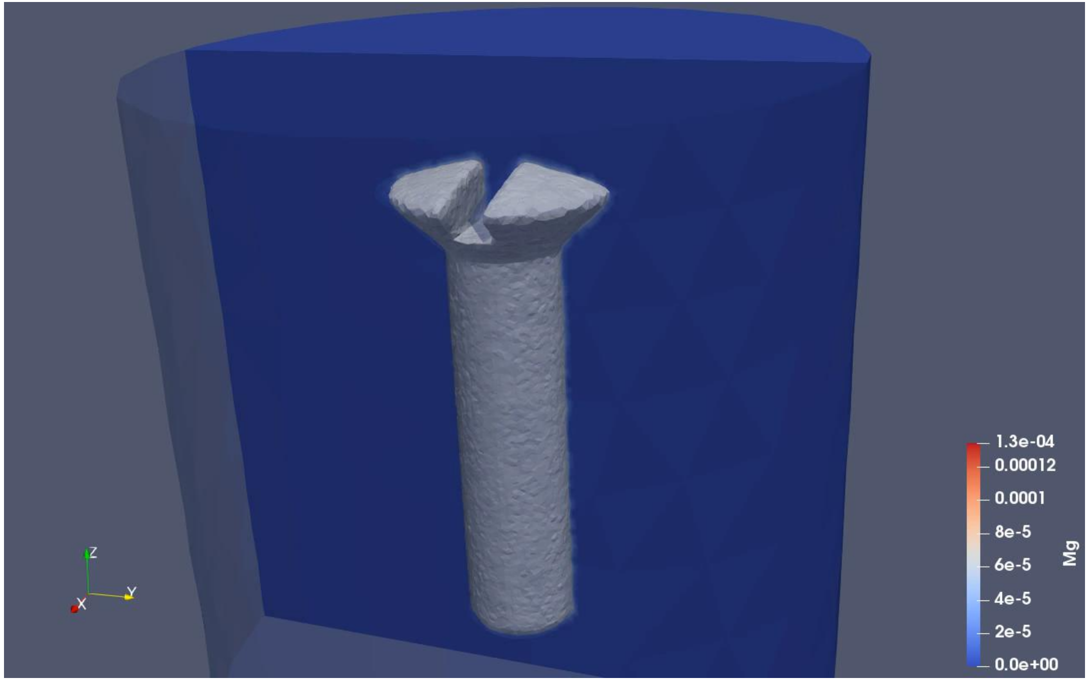
- Not feasible to implement models in sophisticated software packages
- Discretizing PDEs, finite element method
- Level-set for capturing the moving corrosion front
- Adaptively refined mesh generation



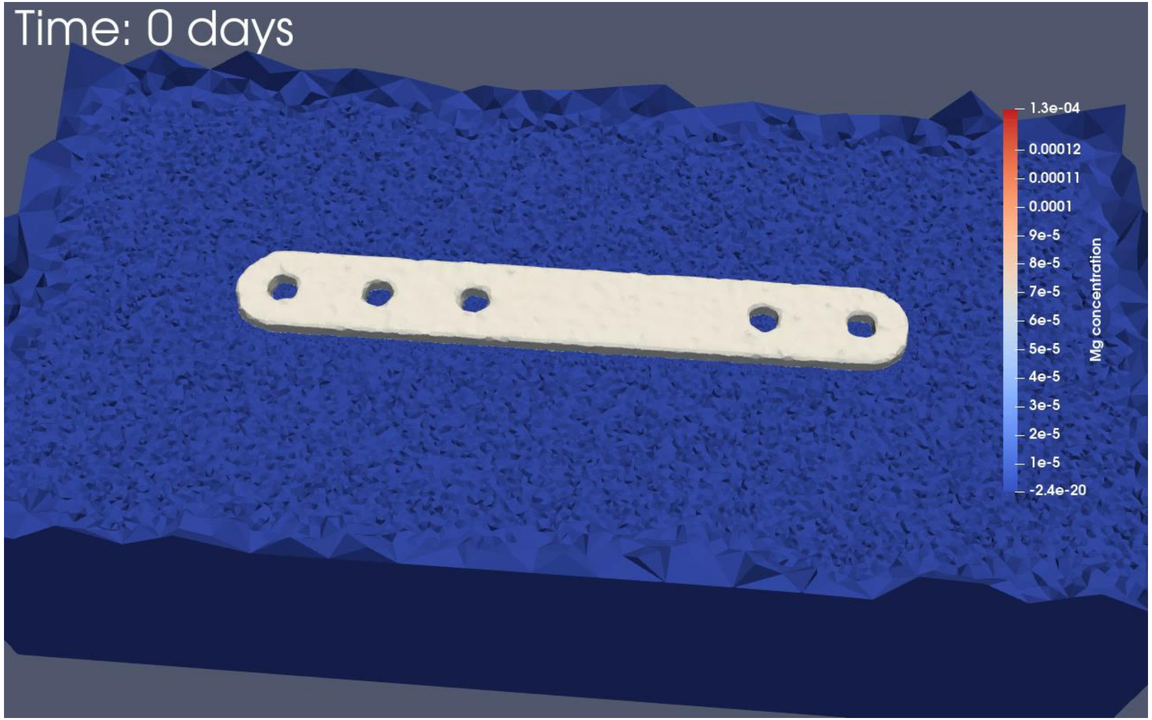
Implementing Computational Model

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

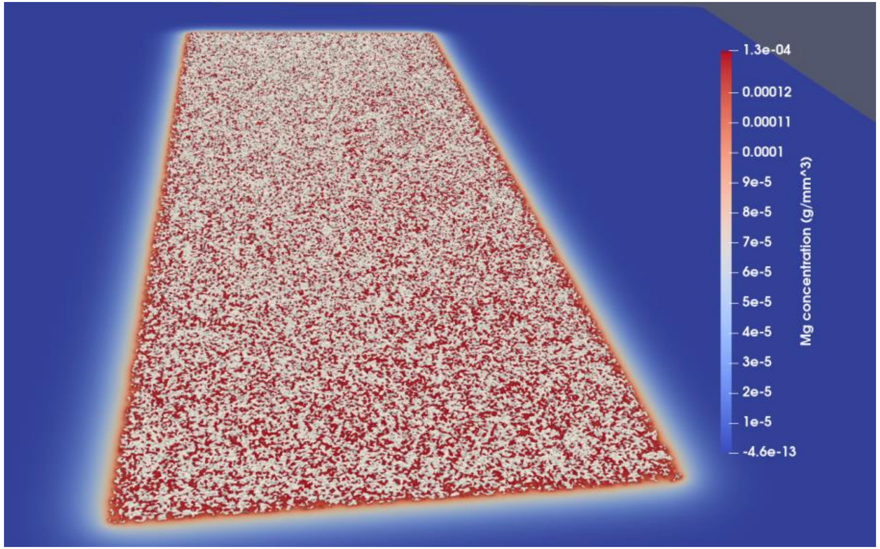
Simple Screw Degradation



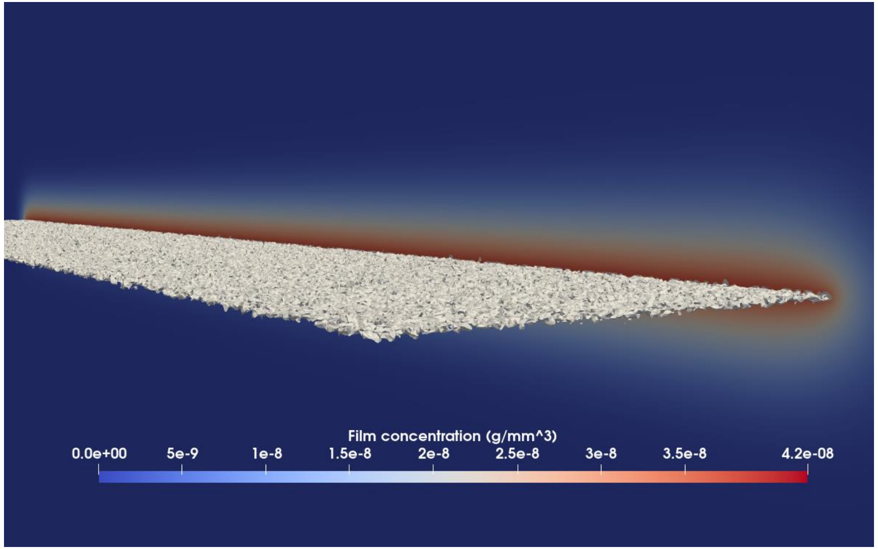
Jaw Bone Plate Degradation



Narrow Cuboid Degradation



Release of Mg ions

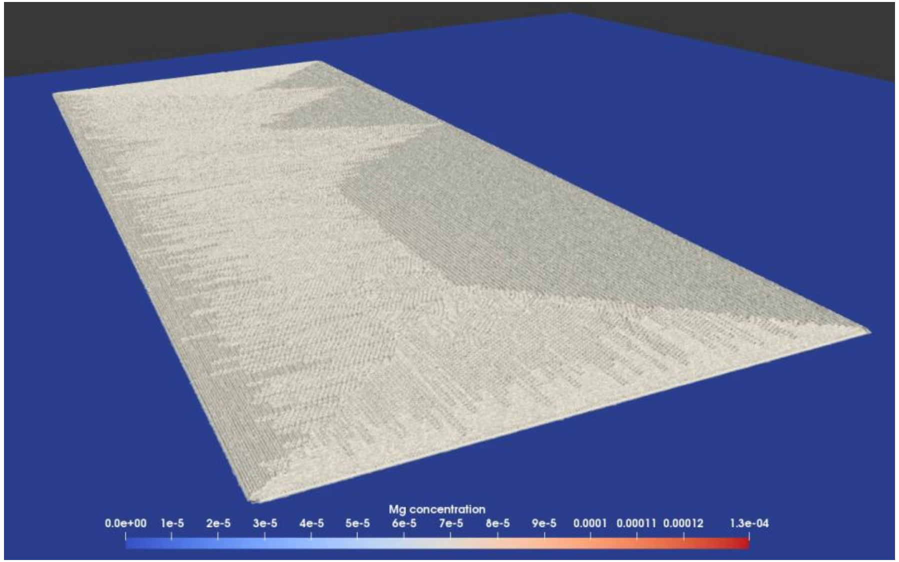


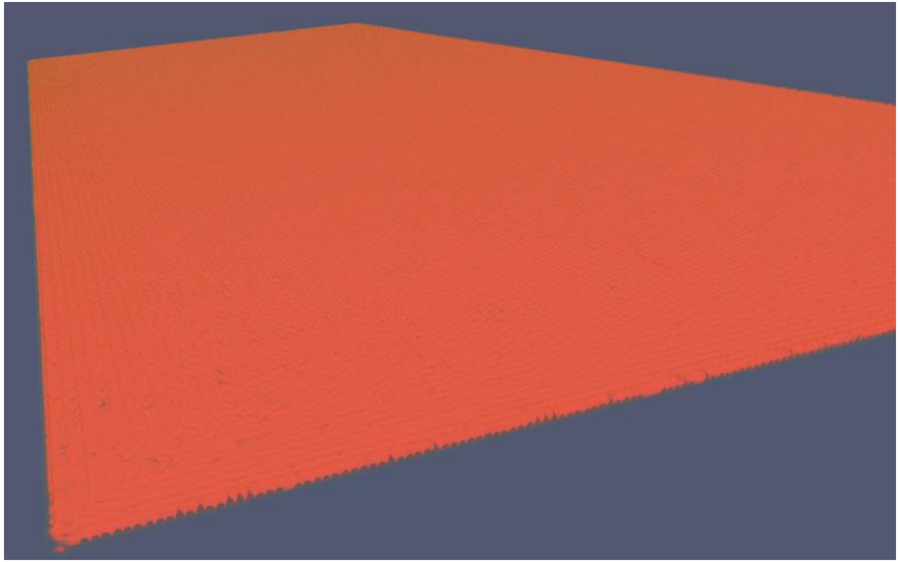
Formation of the protective film

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

KU LEUVEN

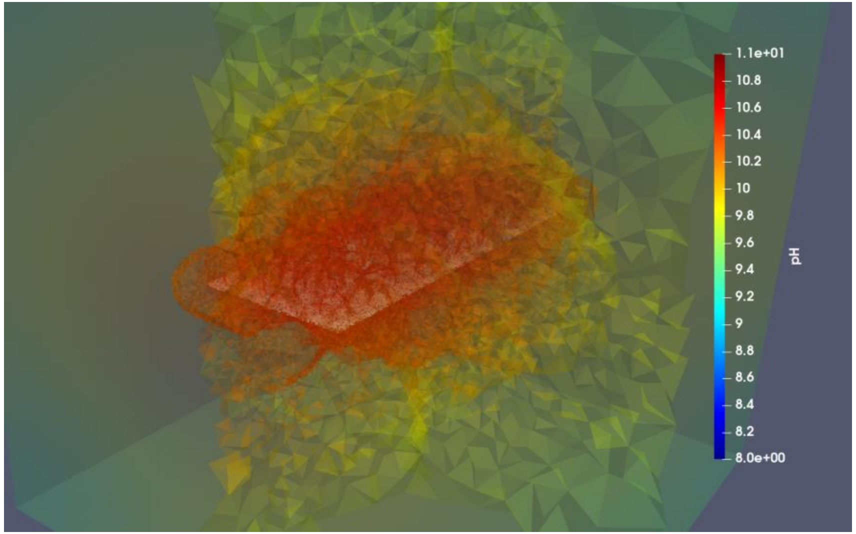
Simulation Results - Degradation



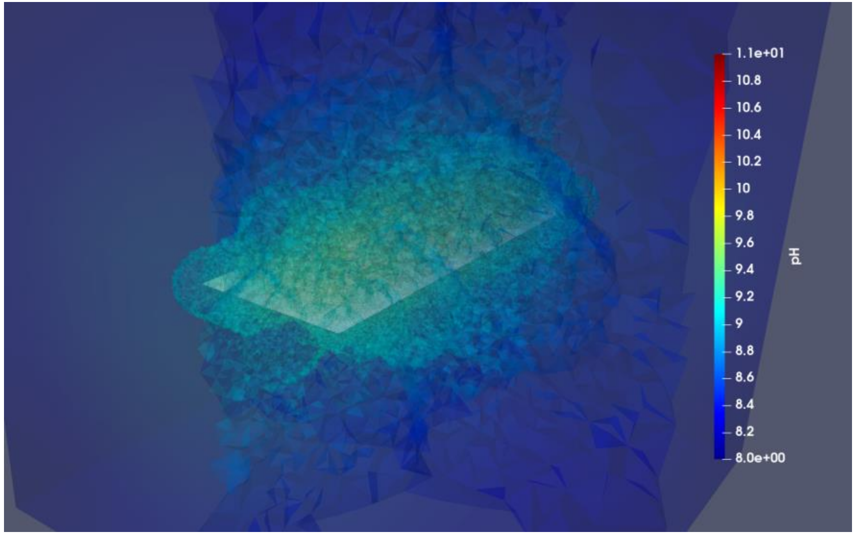


(Barzegari et al., Corrosion Science, 190, 2021)

Simulation Results - pH Change



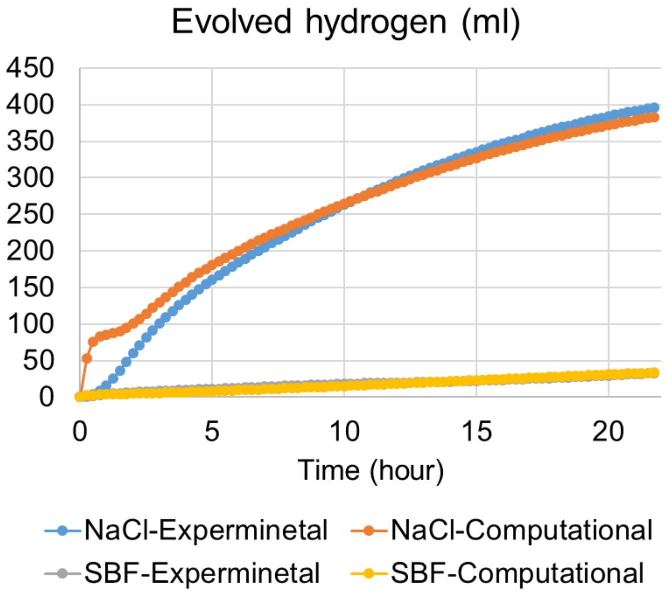
High diffusion (Saline solution)

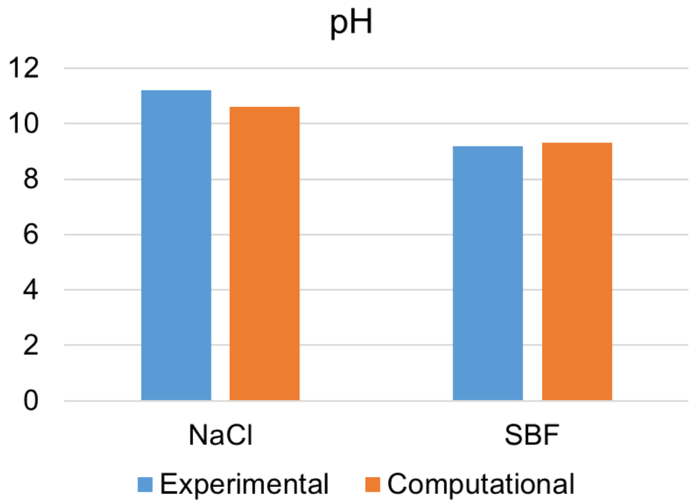


Low diffusion (Buffered solution)

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

Quantitative Results





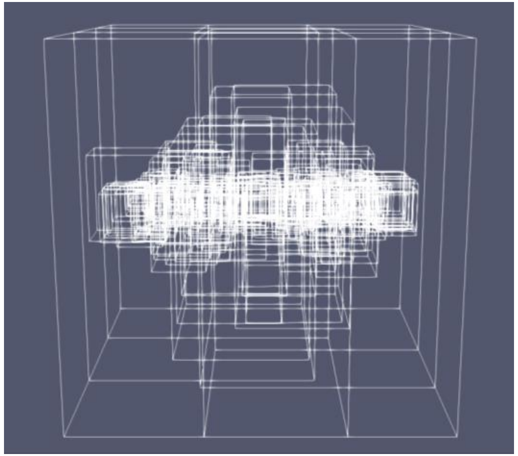
(Barzegari et al., Corrosion Science, 190, 2021)

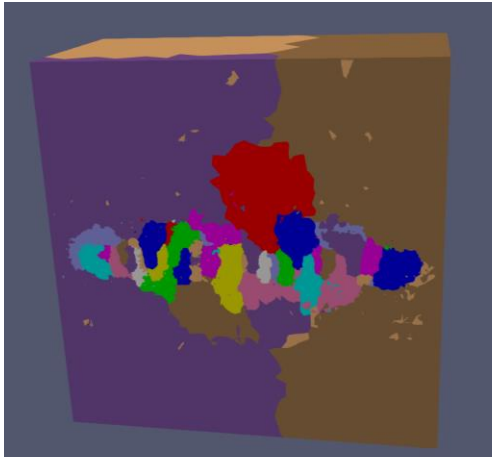
High-Performance Computing Approach

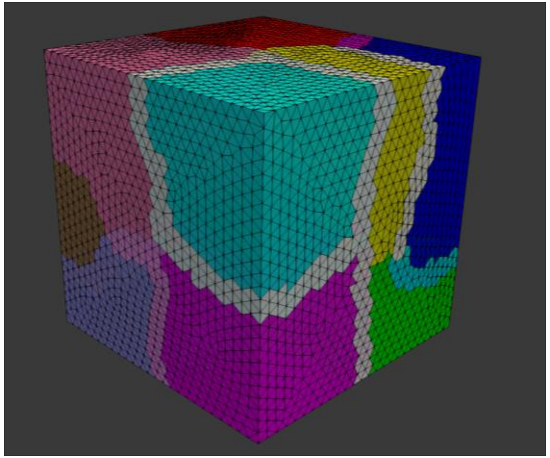
- Distributing the mesh among available resources
  - High-performance mesh decomposition
  - Overlapping Schwarz method
- Solving the linear system of equations
  - BoomerAMG preconditioner
  - GMRES iterative solver
  - Fieldsplit for fluid flow

High-performance Mesh Decomposition

- Mesh is decomposed using overlapping Schwarz preconditioner
- Each partition is assigned to one CPU core (MPI process)

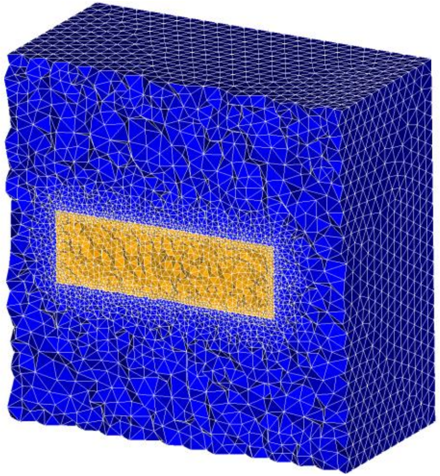


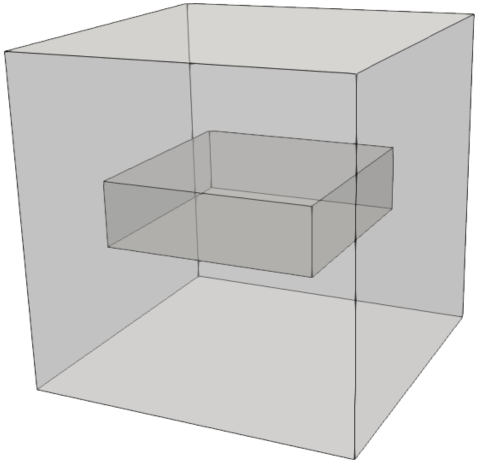




Performance Analysis

- A setup with a thick block
- Only 3 PDEs are solved
- DOF for each PDE ~ 150K
- Elements ~ 900K





(Barzegari et al., International Journal of High Performance Computing Applications, 35, 2021)

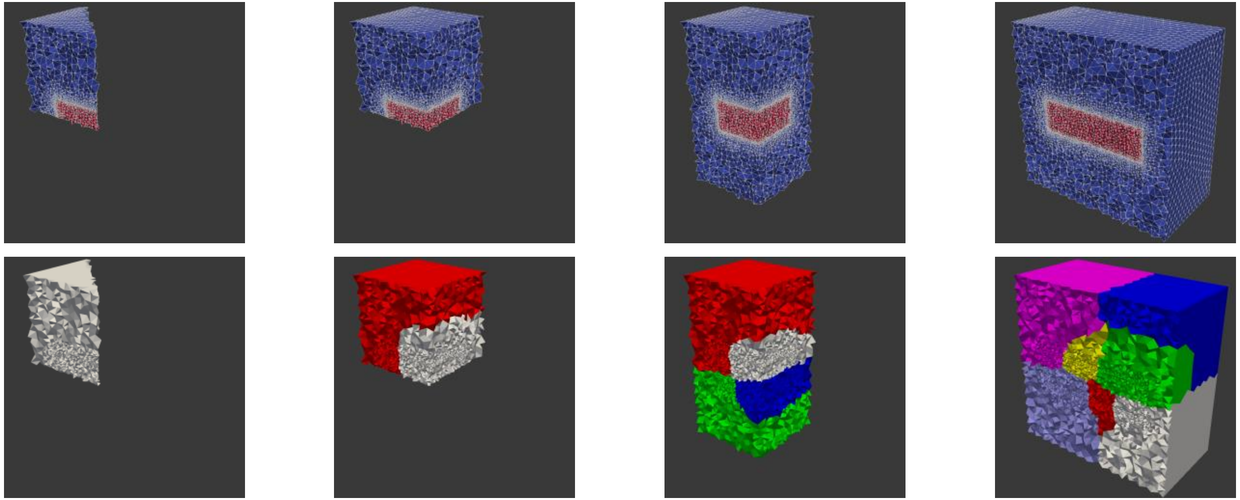
KU LEUVEN

Parallelization Benchmark

- Weak scaling (doubling the problem size while doubling the resources)
- Strong scaling (keeping the problem size constant and doubling the resources)

Problem size

Mesh partitioned to available CPU cores



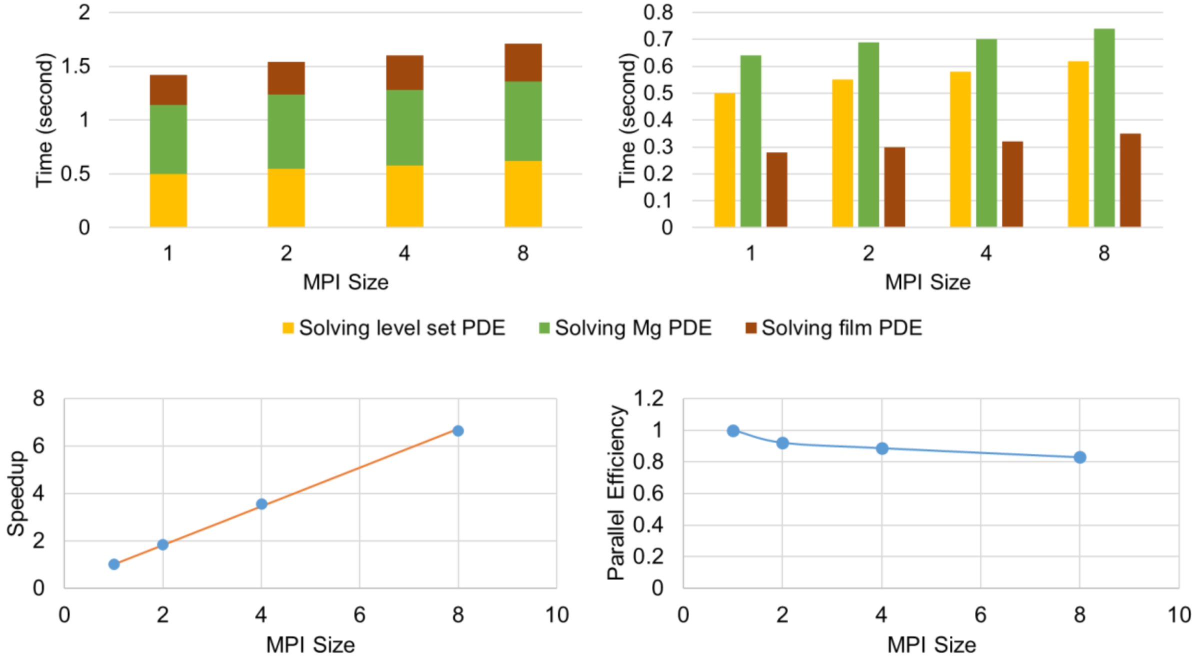
1 core 2 cores 4 cores 8 cores

Weak Scaling Analysis

Gustafson’s law

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

- Sequential part: 18%
- Parallel part: 82%



(Barzegari et al., International Journal of High Performance Computing Applications, 35, 2021)

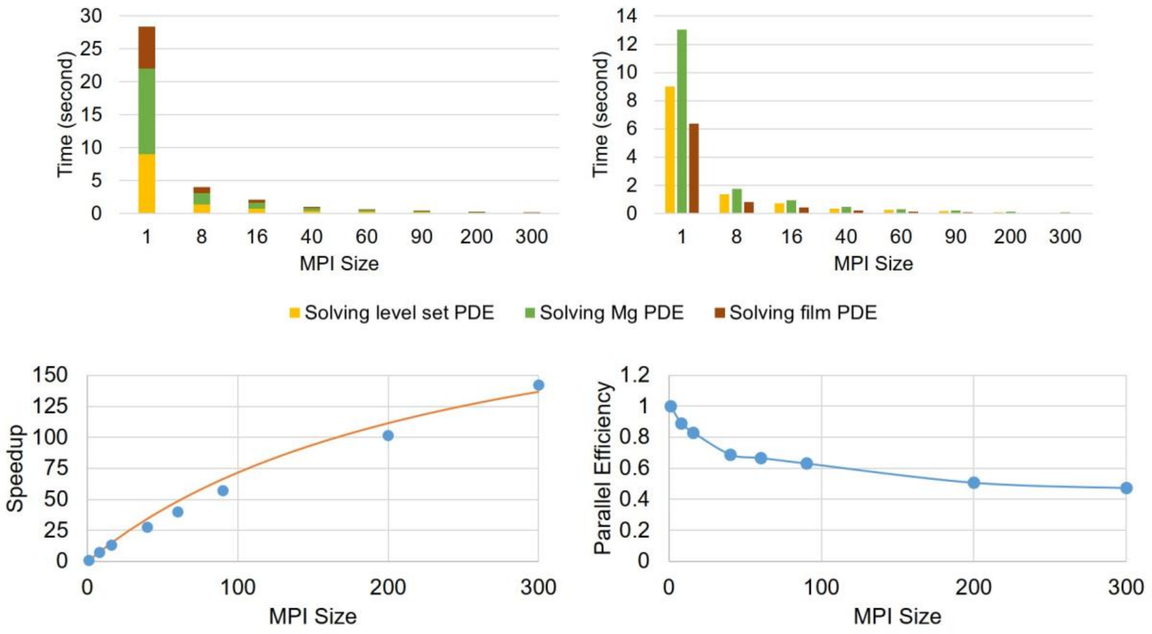
KU LEUVEN

Strong Scaling Analysis

Amdahl's law

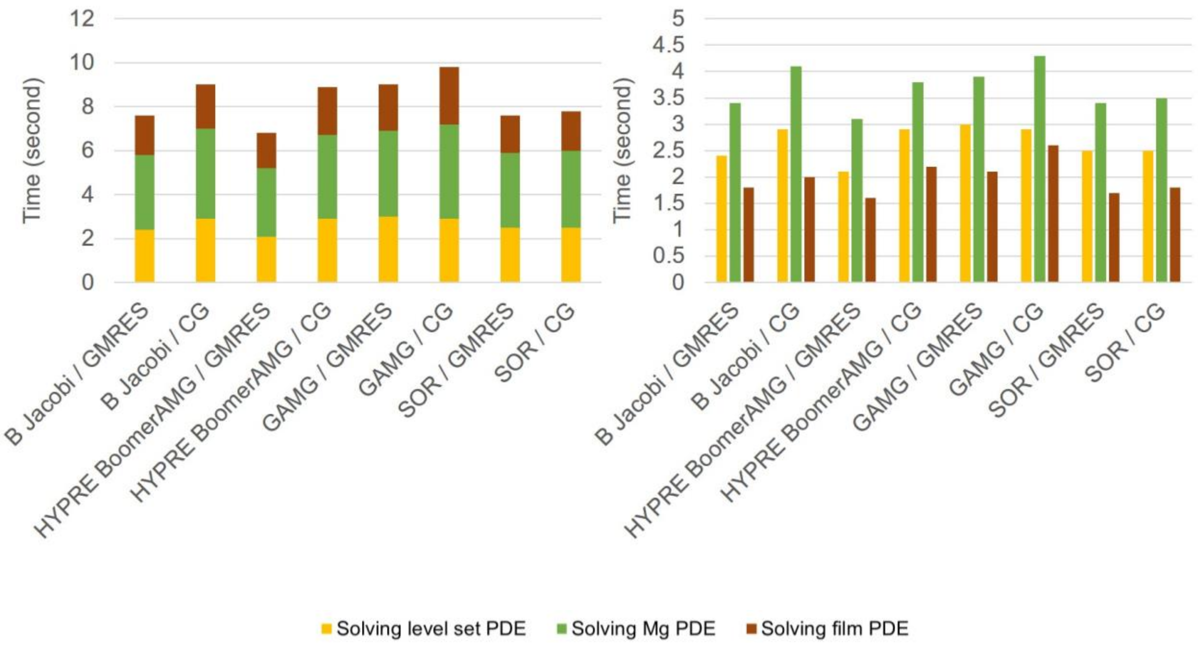
$$\text{Speedup} = \frac{1}{f + \frac{1-f}{N}}$$

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



(Barzegari et al., International Journal of High Performance Computing Applications, 35, 2021)

Preconditioner/Solver Performance



(Barzegari et al., International Journal of High Performance Computing Applications, 35, 2021)

Developed Code & Employed Tools are Open-Source

Mesh Generation
- TetGen
- GMSH
- SALOME
- MMG

CAD Design
- SALOME
- FreeCAD

PDE Solvers
- FreeFEM

Sparse Solvers
- MUMPS
- PETSc (GMRES)

Partitioning
- HPDDM
- ParMETIS

Postprocessing
- ParaView
- Medit
- Seaborn

Parallelization
- OpenMPI
- MPICH

Optimization
- HyperOpt
- Spark

Preconditioning
- HYPRE
- PETSc (fieldsplit)

FE Solvers
- Code Aster

https://github.com/mbarzegary/BioDeg

Conclusion

- The importance of high-performance computing in computational modeling works in the field of biomedical engineering
- Use-case: a quantitative computational model to assess the degradation behavior of biodegradable metallic implants in-silico
- Performance evaluation tests to measure the parallel efficiency and linear scalability of the employed parallelization approaches

Thank you for your attention!

https://mbarzegary.github.io

Mojtaba.Barzegari@kuleuven.be

@MojBarz

@mbarzegary