COMPUTER SIMULATION OF PRESSURE AND VELOCITY FIELDS IN DRUG-ELUTING STENTS USING THE GRIDAP LIBRARY

Rachel Lucena[1] - rachel.lucena@gmail.com
Hyun Ho Shin[2] - hshin@pol.una.py
Gustavo P. Oliveira[3] - gustavo.oliveira@ci.ufpb.br
Gustavo R. Anjos[4] - gustavo.rabello@coppe.ufrj.br
José Pontes[1] - jose.pontes@eng.uerj.br
Sean McGinty[5] - sean.mcginty@glasgow.ac.uk
Norberto Mangiavacchi[1] - norberto@eng.uerj.br
[1]Universidade do Estado do Rio de Janeiro, Departamento de Engenharia Mecânica, RJ, Brazil
[2]Universidad Nacional de Asunción, Núcleo de Investigación y Desarrollo Tecnológico, Paraguay
[3]Federal University of Paraíba, TRIL Lab, João Pessoa, Brazil
[4]Universidade Federal do Rio de Janeiro, COPPE/Department of Mechanical Engineering, RJ, Brazil
[5]University of Glasgow, Division of Biomedical Engineering, Glasgow, UK

Abstract. *Several numerical simulations of drug release from coronary drug-eluting stents were formerly performed by the authors through Matlab© and Octave to understand transport and binding mechanisms over the device/artery interface. As a continued effort, this paper is intended to reproduce results previously obtained of pressure distribution and velocity field, now employing* `Gridap`*, a Julia language-based library, able to explore different approaches of approximation Finite Element spaces, and high performance solution strategies. We compared results obtained employing diverse numerical and computational choices, such as variable mesh refinement levels, finite element interpolation order, pre-conditioner classes, and solver configurations (architecture, memory, multi-threading, etc.). The proposed benchmark problem, even addressing a relatively simple geometry and a linear differential equation, has singular regions that cause numerical difficulties, thus limiting the rate of convergence of the numerical approximations. The main contribution of the paper is to show the benefits of employing Julia as a programming language for scientific applications, and Gridap as a Finite Element library, which provided easy access to high performance resources as multi threading, parallel processing, GPU acceleration, thus great versatility in solving complex problems.*

Keywords: *Finite element method, Numerical simulation, Julia language, Gridap library, Drug-eluting stents*

1. INTRODUCTION

Drug-eluting medical devices constitute an effective approach to drug delivery in the human body, since the treatment is performed directly at the required site (Zilberman et al., 2010). Following this concept, drug-eluting stents are widely used for treatment of coronary diseases. A drug-eluting stent (DES) is a small drug-coated tubular wireframe, used to unclog obstructed coronary arteries. According to Saito and Kobayashi (2021), this device is the most common treatment for patients with acute and chronic coronary artery diseases, mainly because the percutaneous coronary intervention is not invasive.

There are many experimental, clinical and theoretical studies backed by computational approaches regarding drug delivery, effectiveness and efficiency in DESs, such as Hu et al. (2021); Vo et al. (2018); McGinty et al. (2017); McGinty and Pontrelli (2016); Bozsak et al. (2014); Tzafriri et al. (2012); Stone et al. (2009). Of these papers, one verifies that DES fabrication should be constantly improved to reduce risks of late thrombotic events, according to Chisari et al. (2016).

Lucena et al. (2018) studied the dissolution, transport and binding of sirolimus drug on a 2D axisymmetric domain representing the polymer coating layer and the porous artery wall in the vicinity of a stent strut with an in-house computational code. The use of a dedicated Matlab/Octave in-house code allowed to obtain relevant results for the proposed problem. However, the results also showed the need for a more flexible approach to incorporate high performance features required for more complex simulations, such as multi threading, parallel computing, GPU usage, and allow for more choices of Finite Element methods. These features would be very difficult to include in a Matlab/Octave framework, thus a more flexible and extendable framework was deemed required.

Motivated by the previously mentioned mathematical model, in this paper we reproduce the previous results of pressure and velocity fields obtained with Matlab© and Octave, but now employing Julia language and related high-performance libraries: `Gridap.jl`, `GridapGmsh.jl`, `GridapPETSc.jl`, `GridapDistributed.jl`, etc.. The focus is on analysing the effect of numerical and computational choices implemented using the proposed framework, like mesh refinements and finite element types (convergence), pre-conditioners and solvers, implemented on various computer architectures (memory, multi-threading, parallel processing), and other relevant comparisons.

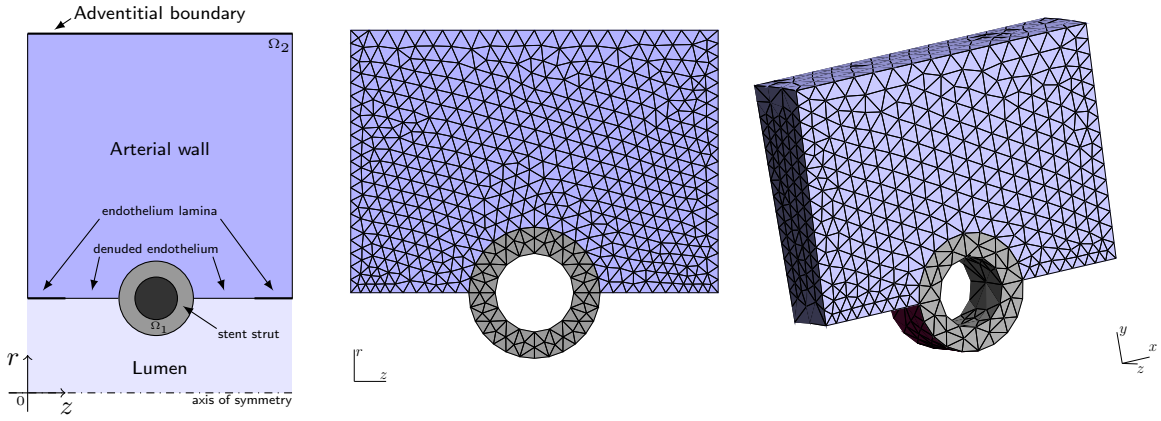
2. METHODOLOGY

2.1 Mathematical model

The arterial wall and the polymer layer are considered as porous media, and flow within these layers is modelled by Darcy's law:

$$\mathbf{u} = -\frac{P_{D_i}}{\mu}\nabla p, \tag{1}$$

where $\mathbf{u}$ is the fluid velocity field, p is the pressure, P_{D_i} is the Darcy permeability of the medium i, where the medium $i = 1$ represents the polymer layer (Ω_1) and the medium $i = 2$ the arterial wall (Ω_2) (see Fig. 1(a)), and μ is the fluid dynamic viscosity. The Eq. (1) is defined on $\Omega = \Omega_1 \cup \Omega_2$.



(a) (b) (c)

Figure 1- (a) Schematic diagram of the numerical domain $\Omega_1 \cup \Omega_2$, where Ω_1 represents the polymer layer and Ω_2 the arterial wall. (b) An example of the bidimensional unstructured triangular mesh employed in the computational simulations. (c) An example of the three-dimensional unstructured mesh employed in the 3D simulations.

The incompressible condition for the fluid velocity is expressed by the continuity equation $\nabla \cdot \mathbf{u} = 0$. Therefore, the Eq. (1) is rewritten as:

$$\nabla \cdot \left(\frac{P_{D_i}}{\mu} \nabla p \right) = 0. \tag{2}$$

The pressure is prescribed on the adventitial boundary and on denuded endothelium (region vicinity of a stent strut) as 0.0 Pa and 9.31×10^3 Pa, respectively. The endothelium lamina is modelled as no-flow in radial direction, *i.e.*, $\mathbf{n} \cdot \nabla p = 0$. These regions are highlighted in a schematic diagram in Fig. 1(a).

The Darcy permeability can be represented by a second order tensor in order to take into account the anisotropy and heterogeneity.

Table 1- Summary of the dimensional parameters employed in the simulations.

Parameter	**Simulated value**
Geometry dimensions	
lumen radius	1.5×10^{-3} m
arterial wall thickness	5.0×10^{-4} m
stent strut diameter	2.5×10^{-4} m
polymer thickness	5.0×10^{-5} m
denuded endothelium length (each piece)	1.5×10^{-4} m
endothelium lamina length (each piece)	7.5×10^{-5} m
domain length in axial direction	7.0×10^{-4} m
Physical parameters	
fluid dynamic viscosity, μ	7.2×10^{-4} Pa· s
polymer layer permeability, P_{D_1}	2.0×10^{-21} m^2
arterial wall permeability, P_{D_2}	2.0×10^{-18} m^2

2.2 Weak formulation

Equation (2) is solved using the Galerkin Finite Element Method (Lucena et al., 2018). The weak form of the Eq. (2) can be expressed as:

$$a(p, q) = 0, \tag{3}$$

where the bilinear form $a(p, q)$ is defined as:

$$a(p, q) \doteq \int_{\Omega} \frac{P_{D_i}}{\mu} \left(\nabla p \cdot \nabla q\right) \mathrm{d}\Omega, \tag{4}$$

where $p \in P$ and $q \in Q_0$, such that $P \subset Q = \{\mathcal{H}^1(\Omega)\}$.

2.3 Numerical method

The Eq. (2) is solved on an unstructured mesh using polynomial base functions and Galerkin Finite Element Method. The finite element simulations are performed in Julia programming language, using the Gridap library. Gridap provides a set of tools for the grid-based approximation of partial differential equations (PDEs) (Verdugo and Badia, 2022; Badia and Verdugo, 2020).

2.4 Numerical experiments

A summary of the information about the meshes used in our numerical experiments for the convergence analysis are available in Tables 2 and 3. The difference between the meshes was determined by a factor for the amount of the mesh points in each boundary of the domain. For the factor equals to one, the separations of the points in each boundary edge range from 0.0083 to 0.017 mm. Higher factor implies more amount of points in each boundary edge. The factor was selected in order that the numbers of nodes and elements are roughly twice between two consecutive grids. The meshes are obtained using Gmsh (Geuzaine and Remacle, 2009).

Table 2- Summary of the two-dimensional grids employed in the mesh convergence analysis.

Id	Factor	Nodes	Elements
`msh2d_1`	0.50	608	1,230
`msh2d_2`	0.75	1,361	2,744
`msh2d_3`	1.00	2,508	5,046
`msh2d_4`	1.50	5,623	11,292
`msh2d_5`	2.00	10,173	20,408
`msh2d_6`	3.00	22,873	45,840
`msh2d_7`	4.00	40,999	82,124
`msh2d_8`	6.00	91,399	182,988
`msh2d_9`	10.00	254,706	509,730

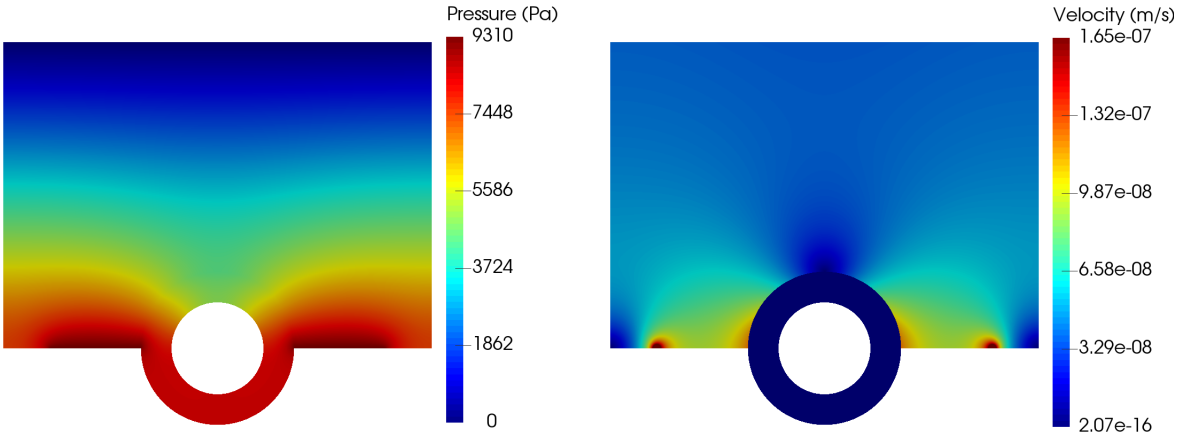
The three-dimensional domain was obtained by extruding the two-dimensional domain around the axis of symmetry by an angle of 5°. The three-dimensional meshes, obtained using Gmsh (Geuzaine and Remacle, 2009), are optimized by NETGEN mesh optimization techniques (Schöberl, 1997).

Table 3- Summary of the three-dimensional grids employed in the mesh convergence analysis.

Id	Factor	Nodes	Elements
msh3d_1	1.0	18,301	101,409
msh3d_2	1.3	35,568	199,740
msh3d_3	1.6	66,165	375,533
msh3d_4	2.0	128,555	736,538
msh3d_5	2.5	238,148	1,373,083
msh3d_6	3.2	487,904	2,833,635
msh3d_7	4.0	943,765	5,506,997

2.5 Pressure and velocity fields

The pressure distribution and velocity field obtained by the two-dimensional simulations are shown in Fig. 2, employing the mesh msh2d_9 and order 2 elements. These results are qualitatively equivalent to Lucena et al. (2018), thus provide a verification of the methodology and the numerical implementation applied.



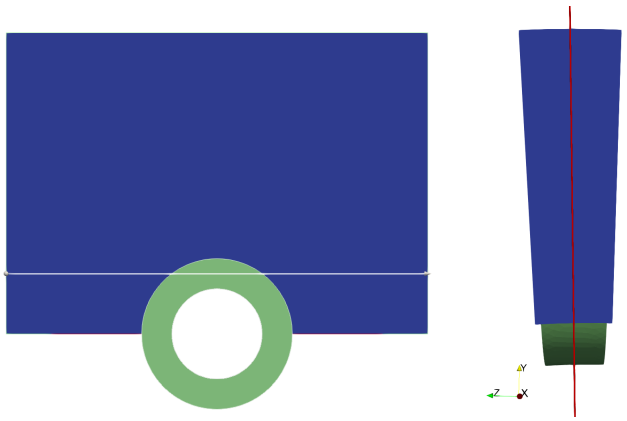
(a) (b)

Figure 2- (a) Pressure and (b) magnitude of the velocity field, from the simulations in two-dimensional approach employing the computational mesh msh2d_9.

Figure 4 shows the pressure and the magnitude of velocities, from the simulations in two-dimensional grids, over a line which crosses both, stent polymer layer and arterial wall. Figure 3(a) depicts a horizontal line over which the pressure and velocities are computed (white colour line). Most part of the line is in arterial wall (blue), and in the central part of the line crosses the stent polymer layer (green). Figure 4(a) shows the pressure, and Figs. 4(b) and 4(c) show the magnitude of the velocities in the arterial wall and stent, respectively. Only the simulation with coarser grid, with Id msh2d_1, presents deviation from the finer grids; and the grids msh2d_5, msh2d_7 and msh2d_9 are one on top of the other. This means that, the refinement of the grid msh2d_5 is adequate for the representation of the pressure and velocity in the region where this line is selected. In Section 2.7 will show that a special refinement is required in the boundary composed by endothelium lamina and denuded endothelium.

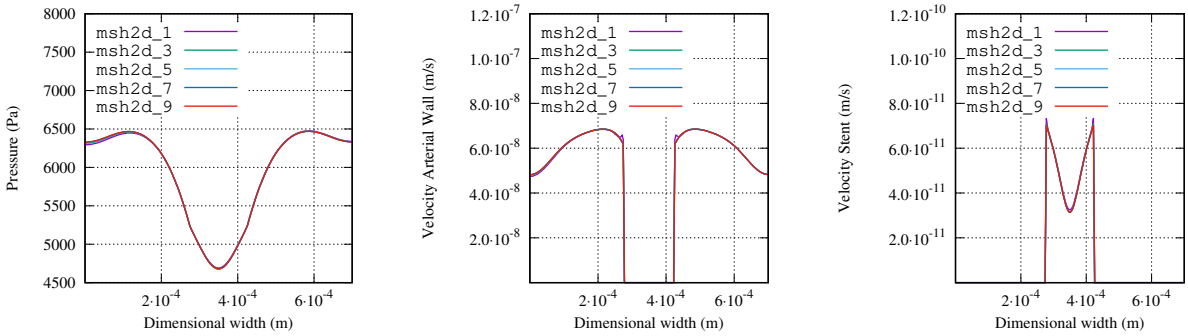
It can be noticed that the velocity in the stent is about three order of magnitude smaller than the velocity in the arterial wall. This is due to the differences in the permeabilities, where in the stent polymer layer has lower permeability compared to the value in the arterial wall.

Results obtained by the 2D (axis-symmetric) and 3D simulations (see Fig. 5) are equivalent for corresponding meshes, thus providing a verification of the implementation.



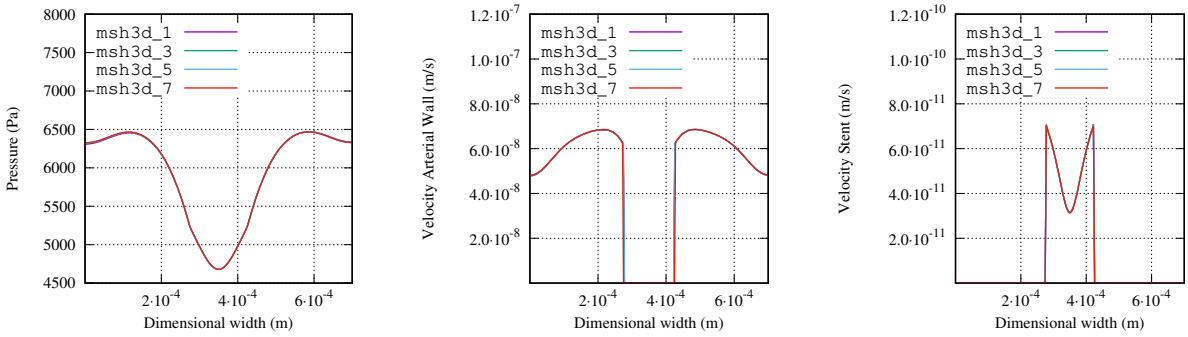
(a) (b)

Figure 3- Position of the horizontal line segment where the pressure and velocities are computed: (a) Position representation of the horizontal line in two-dimensional simulations. (b) The lateral view of the three-dimensional domain employed in 3D simulations along the symmetry-axis, of the position of a plane corresponding to a two-dimensional simulation. The 2D domain represents a plane indicated in red line in an angle of 2.5°.



(a) (b) (c)

Figure 4- Pressure and magnitude of velocity over a line in simulations with five two-dimensional grids: (a) pressure, (b) velocity in the arterial wall and (c) velocity in the polymer stent.



(a) (b) (c)

Figure 5- Pressure and magnitude of velocity over a line in simulations with four three-dimensional grids: (a) pressure, (b) velocity in the arterial wall and (c) velocity in the polymer stent.

2.6 Volume flux convergence analysis

The volume flux balance can be employed as a verification and estimate of the approximation errors. The volume fluxes at the adventitial boundary and at the lumen boundary are computed and compared. The results obtained are shown in Table 4.

The volume fluxes are computed by the Eq. (5):

$$\dot{q_i} = \int_{\Gamma_i} \mathbf{n} \cdot \mathbf{u} \, \mathrm{d}\Gamma, \tag{5}$$

where $i = 1, 2$ and 3, with $\Gamma_1 = \Gamma_a$ is adventitial boundary, $\Gamma_2 = \Gamma_d$ is the denuded endothelium boundary and $\Gamma_3 = \Gamma_b$ is the stent strut bottom boundary.

Table 4- Volume flux convergence analysis in two and three-dimensional simulations.

mesh Id	$\dot{q_1}$	$\dot{q_2}$	$\dot{q_3}$	balance	Relative error (%)
msh2d_1	5.158×10^{-14}	-4.928×10^{-14}	-5.156×10^{-18}	2.303×10^{-15}	2.28
msh2d_2	5.146×10^{-14}	-4.957×10^{-14}	-3.135×10^{-18}	1.888×10^{-15}	1.87
msh2d_3	5.139×10^{-14}	-4.981×10^{-14}	-2.914×10^{-18}	1.577×10^{-15}	1.56
msh2d_4	5.133×10^{-14}	-5.011×10^{-14}	-2.273×10^{-18}	1.214×10^{-15}	1.20
msh2d_5	5.130×10^{-14}	-5.032×10^{-14}	-1.872×10^{-18}	9.734×10^{-16}	0.96
msh2d_6	5.127×10^{-14}	-5.049×10^{-14}	-1.458×10^{-18}	7.827×10^{-16}	0.77
msh2d_7	5.126×10^{-14}	-5.060×10^{-14}	-1.249×10^{-18}	6.542×10^{-16}	0.64
msh2d_8	5.124×10^{-14}	-5.073×10^{-14}	-1.156×10^{-18}	5.104×10^{-16}	0.50
msh2d_9	5.123×10^{-14}	-5.093×10^{-14}	-9.214×10^{-19}	2.927×10^{-16}	0.29
msh3d_1	4.474×10^{-15}	-4.278×10^{-15}	-2.214×10^{-19}	1.951×10^{-16}	2.23
msh3d_2	4.472×10^{-15}	-4.296×10^{-15}	-2.578×10^{-19}	1.762×10^{-16}	2.01
msh3d_3	4.473×10^{-15}	-4.313×10^{-15}	-2.156×10^{-19}	1.603×10^{-16}	1.84
msh3d_4	4.473×10^{-15}	-4.314×10^{-15}	-1.822×10^{-19}	1.587×10^{-16}	1.81
msh3d_5	4.472×10^{-15}	-4.333×10^{-15}	-1.703×10^{-19}	1.384×10^{-16}	1.57
msh3d_6	4.471×10^{-15}	-4.348×10^{-15}	-1.345×10^{-19}	1.230×10^{-16}	1.39
msh3d_7	4.470×10^{-15}	-4.369×10^{-15}	-1.310×10^{-19}	1.015×10^{-16}	1.14

An estimate of the asymptotic order of convergence is given by

$$N = -\frac{\log\left(\frac{E_F}{E_C}\right)}{\log\left(\frac{h_F}{h_C}\right)}, \tag{6}$$

where E_C and E_F are the absolute errors obtained on a coarse and a fine mesh, while h_F and h_C are the representative mesh lengths in the two meshes. Taking msh2d_9 as the fine mesh, and msh2d_1, we obtain $N = 0.68$, thus the volume flow convergence is sub-linear. Similarly, taking msh3d_7 and msh3d_1 for three-dimensional grids, $N = 0.50$ is obtained for volume flow convergence.

2.7 Singularity convergence analysis

The abrupt change on the boundary condition at the endothelial boundary, from the denuded endothelial region to the impermeable lamina, results in a flow which is similar to the complex potential flow of the form

$$F(z) = Az^n, \tag{7}$$

where the components of the velocity field is given by

$$v_r = \quad nAr^{n-1}\cos(n\theta) \qquad \text{and} \qquad v_\theta = -nAr^{n-1}\sin(n\theta). \tag{8}$$

In Cartesian coordinates, it can be written as

$$v_x = v_r\cos\theta - v_\theta\sin\theta \qquad \text{and} \qquad v_y = v_r\sin\theta + v_\theta\cos\theta. \tag{9}$$

The case $n = 1/2$ represents the flow on a wedge with an angle $\alpha = 2\pi$, i.e., the wedge flow around a sharp edge, that closely represents the flow in the vicinity of the singularity. A schematic draw of the flow is shown in Fig. 6.

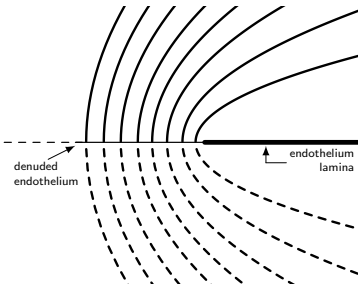
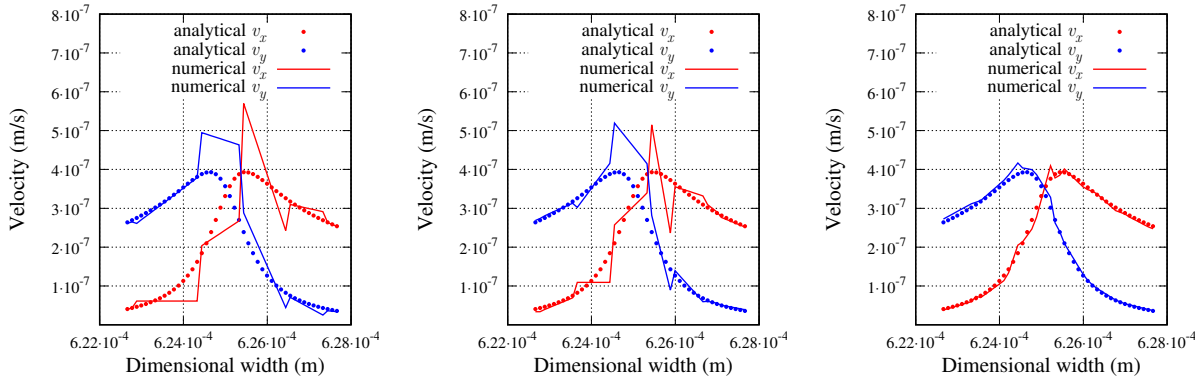


Figure 6- Sketch of flow on a wedge with an angle $\alpha = 2\pi$.

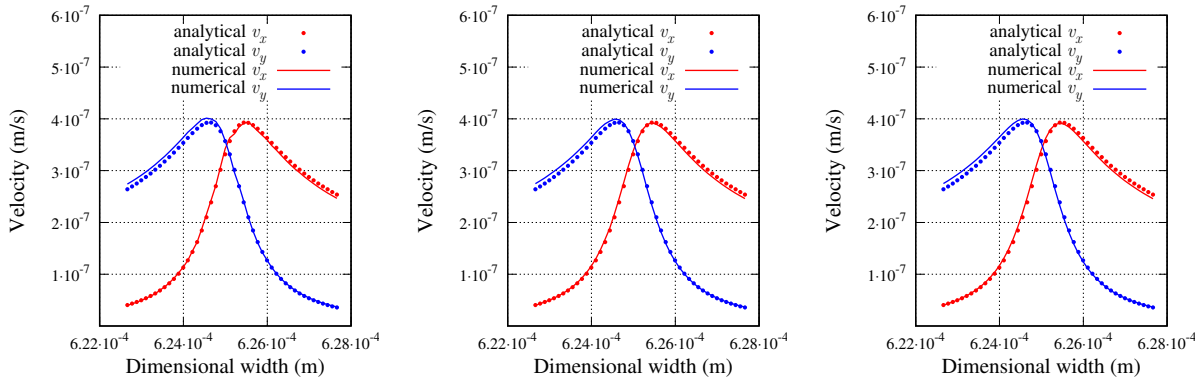
The convergence of the solution close to the singularity is slow, and even employing a very fine mesh the velocity field is still not very accurate when employing low order elements. Figure 7 compares the solution obtained for the velocity components near the singularity employing order 2 elements and various mesh refinements, showing the slow convergence of the solution as the mesh is refined with this kind of elements.



(a) `msh2d_7` (b) `msh2d_8` (c) `msh2d_9`

Figure 7- Comparison between analytical and numerical solutions for the velocity components obtained by Eqs. (9) and Darcy's law (Eq. (1)), respectively.

Better accuracy is obtained by increasing the order of the elements, such as employing polynomial base functions of order 3, 4 and 5, as can be seen in Fig. 8.



(a) order 3 (b) order 4 (c) order 5

Figure 8- Comparison of the velocity components between analytical and numerical solutions close to the singularity for the most refined mesh (`msh2d_9`) employing polynomial base functions of order 3, 4 and 5.

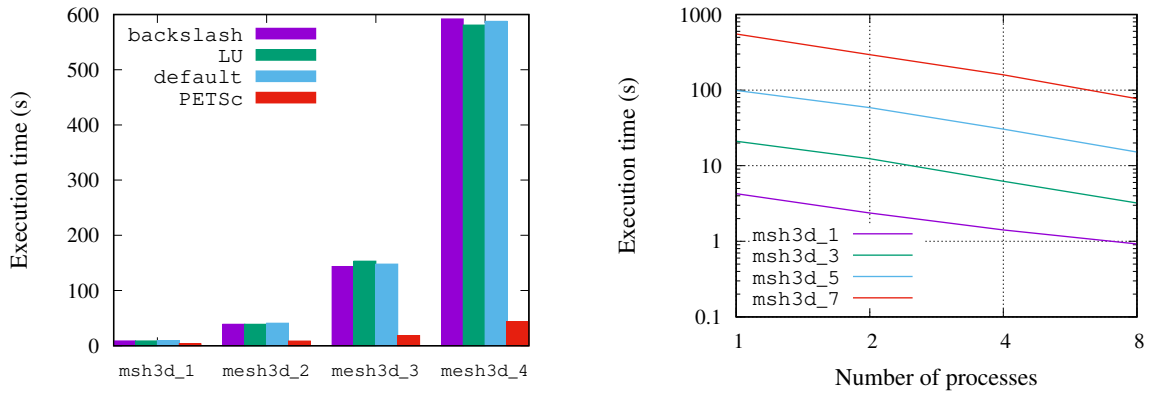
2.8 Computational cost and performance

Several test were performed to asses the computational performance of the implemented methods. The Gridap library allows the usage of various solver types (Backslash, LU, default, GAMG PETSc). The proposed problem setup provides a useful benchmark to compare various alternatives, and the effect of the employed hardware.

Figure 9(a) shows a comparison of performance employing various solver libraries. There is no appreciable difference between the Backslash, LU and Gridap default solvers. Furthermore, the execution time and memory usage increase in these solvers considerably with mesh refinement. On the other hand, the linear solver of PETSc showed very good performance in both, the execution time and memory usage.

The effect of the CPU architecture was analysed employing a cluster. Results were obtained using the following configurations: 4 nodes with two Xeon 2.6 Ghz (16 cores), 128 Gb RAM, GPU Nvidia Tesla V100, Dell 740.

For a single node, multi threading performance was verified varying the number of processes per node, from 1 to 8, with results shown in Fig. 9(b). The implementation showed very good speed-up, even for small 3D problems. A comparison with the previous Octave implementation results using `msh2d_5` (Lucena et al., 2018) show a decrease of $90\times$ on the execution time.



(a) (b)

Figure 9- Execution time: (a) performance for various solver libraries and (b) speed-up varying the number of processes using three-dimensional grids.

3. CONCLUSIONS

This paper addresses, as a benchmark problem, the computation of the pressure and velocity fields in the vicinity of a coronary stent by the Finite Element Method, reproducing results previously obtained by Lucena et al. (2018), now exploring mesh refinement and high performance solution strategies on accuracy and computational cost. The addressed problem is a challenging benchmark since, even being defined by a relatively simple geometry and a linear differential equation, has singular regions that cause numerical difficulties, thus limiting the rate of convergence of the numerical approximations. The main contribution of the paper was to show the potentiality of employing Julia as a scientific language, and Gridap as a Finite Element library, which provided easy access to high performance resources as multi threading, parallel processing, GPU acceleration, thus great versatility in solving complex problems.

Acknowledgements

The authors would like to acknowledge financial support from FAPERJ, CAPES, CNPq, and the University of Glasgow EPSRC GCF ISF fund.

REFERENCES

Badia, S. and Verdugo, F. (2020), Gridap: An extensible finite element toolbox in Julia. Journal of Open Source Software, 5, 2520.

Bozsak, F.; Chomaz, J. and Barakat, A. (2014), Modelling the transport drugs of eluted from stents: physical phenomena driving drug distribution in the arterial wall. Biomech Model Mechanobiol, 13, 327–347.

Chisari, A.; Pistritto, A.; Piccolo, R.; Manna, A.L. and Danzi, G. (2016), The ultimaster biodegradable-polymer sirolimus-eluting stent: An updated review of clinical evidence. Int. J. Mol. Sci., 17(9), 1490.

Geuzaine, C. and Remacle, J.F. (2009), Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International journal for numerical methods in engineering, 79, 1309–1331.

Hu, S.; Li, Z.; Shen, D.; Zhu, D.; Huang, K.; Su, T.; Dinh, P.U.; Cores, J. and Cheng, K. (2021), Exosome-eluting stents for vascular healing after ischaemic injury. Nature biomedical engineering, 1–15.

Lucena, R.; Mangiavacchi, N.; Pontes, J.; Anjos, G. and McGinty, S. (2018), On the transport through polymer layer and porous arterial wall in drug-eluting stents. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40.

McGinty, S.; King, D. and Pontrelli, G. (2017), Mathematical modelling of variable porosity coatings for controlled drug release. Medical engineering & physics, 45, 51–60.

McGinty, S. and Pontrelli, G. (2016), On the role of specific drug binding in modelling arterial eluting stents. J Math Chem, 54, 967–976.

Saito, Y. and Kobayashi, Y. (2021), Contemporary coronary drug-eluting and coated stents: A mini-review. Cardiovascular intervention and therapeutics, 36, 20–22.

Schöberl, J. (1997), NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. Computing and Visualization in Science, 1, 41–52.

Stone, G.W.; Lansky, A.J.; Pocock, S.J.; Gersh, B.J.; Dangas, G.; Wong, S.C.; Witzenbichler, B.; Guagliumi, G.; Peruga, J.Z.; Brodie, B.R., et al. (2009), Paclitaxel-eluting stents versus bare-metal stents in acute myocardial infarction. New England Journal of Medicine, 360, 1946–1959.

Tzafriri, A.R.; Groothuis, A.; Price, G.S. and Edelman, E.R. (2012), Stent elution rate determines drug deposition and receptor-mediated effects. Journal of Controlled Release, 161, 918–926.

Verdugo, F. and Badia, S. (2022), The software design of gridap: A finite element package based on the julia JIT compiler. Computer Physics Communications, 276, 108341.

Vo, T.T.; Morgan, S.; McCormick, C.; McGinty, S.; McKee, S. and Meere, M. (2018), Modelling drug release from polymer-free coronary stents with microporous surfaces. International journal of pharmaceutics, 544, 392–401.

Zilberman, M.; Kraitzer, A.; Grinberg, O. and Elsner, J.J. (2010), Drug-eluting medical implants. Drug Delivery, 299–341.