Received: 6 November 2022 | Revised: 5 April 2023 | Accepted: 27 April 2023

DOI: 10.1002/cnm.3731

SPECIAL ISSUE ARTICLE

WILEY

Dynamic adaptive moving mesh finite-volume method for the blood flow and coagulation modeling

Kirill M. Terekhov[1,2] | **Ivan D. Butakov**[2] | **Alexander A. Danilov**[1,2,3] | **Yuri V. Vassilevski**[1,2,3]

[1]Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russia

[2]Sirius University of Science and Technology, Sochi, Russia

[3]Sechenov University, Moscow, Russia

Correspondence

Kirill M. Terekhov, Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow 119333, Russia.
Email: terekhov@inm.ras.ru

Present address

Kirill M. Terekhov, Moscow, Russia.

Funding information

Ministry of Education and Science of the Russian Federation, Grant/Award Number: 075-15-2022-286; Russian Science Foundation, Grant/Award Number: 21-71-20024

Abstract

In this work, we develop numerical methods for the solution of blood flow and coagulation on dynamic adaptive moving meshes. We consider the blood flow as a flow of incompressible Newtonian fluid governed by the Navier–Stokes equations. The blood coagulation is introduced through the additional Darcy term, with a permeability coefficient dependent on reactions. To this end, we introduce moving mesh collocated finite-volume methods for the Navier–Stokes equations, advection–diffusion equations, and a method for the stiff cascade of reactions. A monolithic nonlinear system is solved to advance the solution in time. The finite volume method for the Navier–Stokes equations features collocated arrangement of pressure and velocity unknowns and a coupled momentum and mass flux. The method is conservative and inf-sup stable despite the saddle point nature of the system. It is verified on a series of analytical problems and applied to the blood flow problem in the deforming domain of the right ventricle, reconstructed from a time series of computed tomography scans. At last, we demonstrate the ability to model the coagulation process in deforming microfluidic capillaries.

KEYWORDS

adaptive mesh, coagulation, collocated, finite volume, inf-sup stability, moving mesh

1 | INTRODUCTION

This work is focused on robust numerical methods for the blood flow and coagulation[1] in a deforming vascular system. The blood coagulation cascade is a stiff system that has a threshold response to model parameters.[2] This issue limits the time step size for the fully implicit and stable integration of the coupled model. This restriction leads to an impractical total computing time for three-dimensional problems of practical interest. To allow for a larger time step size, we extend a fully implicit finite-volume method[3] for moving meshes and combine it with the matrix weighted Euler method for stiff reaction systems.[4] To capture the peculiarities of the flow, we consider dynamic mesh adaptation along the simulation. This allows us to robustly model blood flow and coagulation in a deforming domain. The mathematical model for the blood coagulation model was developed through a series of works by Bouchnita et al.[5–7]

The typical problem of the collocated methods is the inf-sup instability issue.[8–10] A usual solution to the problem is to use a staggered velocity arrangement.[11–13] However, with the staggered scheme, it is hard to retain the conservation properties on unstructured meshes.[14,15] Another solution is to use the Rhie-Chow interpolation method[16] with the collocated arrangement of unknowns. It is standard in industrial applications such as CD-Adapco STAR-CCM,[17] Ansys

https://doi.org/10.1002/cnm.3731

CFX,[18] Ansys Fluent,[19] Converge CFD.[20] It is also employed in widely used OpenFOAM package using implicit fully coupled approach[21] and later extended to moving meshes.[22] Bouchnita et al.[7] implemented a similar coagulation model using OpenFOAM.

The present work extends the collocated finite-volume method considered earlier in References [23–25] to handle moving meshes. It realizes a general concept of stable flux discretization of saddle-point systems for vectors of several unknowns[3] using a combination of harmonic averaging point concept[26] and a flux-difference splitting method[27–30] for one-sided fluxes with positive matrix coefficients.[31,32] Single-sided fluxes are required to be linearity preserving,[33] that is, exact on linear solutions. The harmonic averaging point is the vector of unknowns at the interface, obtained through the equality of single-sided fluxes. The approach was first applied to the anisoptropic heterogeneous mixed Darcy formulation in Reference [34], leading to first-order accurate monotone method but violating linearity preservation. The method was further extended to mechanics,[35,36] poroelasticity,[37–41] and incompressible fluid flow.[23–25] These works demonstrated that the inf-sup stability issue[8] does not affect the proposed collocated finite-volume methods.

The treatment of various types of the boundary conditions is an integral part of the method. Various approaches have been suggested to address absence of pressure boundary condition for pressure,[42–46] among which we use the combination of available boundary conditions and momentum equation.

There are many works on moving mesh finite volume methods.[47–54] In this work, we adopt the space–time finite volume method that considers the moving mesh as a static four-dimensional mesh.[55–61] The method is conservative by construction. This approach has the advantage of avoiding interpolation between old and new grids, which is necessary in the static methods. We assume that mesh tangling does not occur, which otherwise requires mesh untangling and conservative solution remapping.[62,63]

Dynamic mesh adaptation provides substantial gains by reducing time needed to obtain the solution.[64–68] It requires proper adaptation criteria to maintain accuracy.[69–72] In this work we use hierarchical adaptive mesh refinement or H-refinement on general meshes.[73–81] It does not require complex conservative solution remapping since the new mesh cells are embedded into old cells, but leads to "hanging nodes" issue. Some methods suffer accuracy loss or unphysical oscillations at the interface of coarse and fine meshes.[13,82]

The coagulation cascade is a stiff reaction system. A large body of work on stiff problems is dedicated to the choice and stability of numerical schemes.[83–87] In this work, we combine the forward and backward Euler methods with a matrix weight, chosen at each nonlinear iteration to reproduce the solution of an exponential integrator.[4] The method allows us to perform large time steps.

The numerical implementation of the mathematical model is built on top of INMOST, an open-source library.[88–90] It provides tools for complex parallel mesh modification and balancing,[80,91] as well as tools for linear system assembly using automatic differentiation and for linear system solution. There are other mesh libraries allowing for parallel mesh modification, such as Dune,[92] project DuMuX,[93,94] STK mesh from Trilinos package,[95] OpenFOAM.[96] Packages for parallel mesh management are MOAB[97] and MSTK.[98] In this work, we are concerned mostly with practical aspects of the solution of the Navier–Stokes problem and its combination with a system of advection-reaction–diffusion equations on dynamic adaptive moving meshes using INMOST functionality.

The article is organized as follows. In Section 3, we formulate and derive the moving mesh finite volume method for the Navier–Stokes system. In Section 4, we consider evaluation of four-dimensional geometry. In Section 5, we discuss the solution of the resulting nonlinear problem and mesh adaptation. The verification tests for the method are performed in Section 6, followed by the simulation of blood flow in the right ventricle in Section 7. The system for the blood clotting factors is introduced in Section 8 along with the approximation methods of the reaction system and the moving finite-volume method for the convection-diffusion problem. In Section 9, we address the solution of the nonlinear problem of blood flow and coagulation. Finally, in Section 10 we consider the impact of movement on the embolization of normal pooled plasma in microfluidic capillaries with a patch corresponding to damaged endothelium.

2 | MATERIALS AND METHODS

All the numerical methods used to conduct the numerical experiments are presented in this work in Sections 3, 4, 5 for solving the Navier–Stokes equations, and Sections 8, 9 for solving the blood coagulation process. The analytical tests used to verify the method for the Navier–Stokes equations are fully described in Section 6. The moving mesh of the right ventricle for the numerical experiment in Section 7 was obtained from the authors of Reference [99]. The consent of the patient for the data processing was obtained through Sechenov University. The computational domain, problem setup, and model parameters for the numerical experiment of blood coagulation in Section 10 were obtained from the authors of Reference [1] and the rules for the domain deformation are described in Section 10. The numerical methods were implemented using the open-source INMOST platform[100] providing adaptive moving mesh management.

3 | NAVIER–STOKES SYSTEM AND FINITE-VOLUME METHOD

We consider the Navier–Stokes system of equations in a deforming domain $\Omega(t)$:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I}) = \mathbf{f} & \\ \qquad \operatorname{div}(\mathbf{u}) = 0 & \text{in} \quad \Omega(t), \\ \boldsymbol{\alpha}(\mathbf{u}-\mathbf{w}) + \boldsymbol{\beta}(\boldsymbol{\tau}(\mathbf{u}) - p\mathbb{I})\mathbf{n} = \mathbf{r} & \text{on} \quad \partial\Omega(t). \end{cases} \tag{1}$$

The domain $\Omega(t)$ smoothly changes with time, $\mathbf{u} = [u, v, w]^T$ is the velocity vector, where $u, v, w \in H^1(\Omega)$, and $p \in L^2(\Omega)$ is the pressure field, $\boldsymbol{\tau}(\mathbf{u})$ is the stress tensor with constant kinematic viscosity $\nu = \text{const}$:

$$\boldsymbol{\tau}(\mathbf{u}) = 2\nu\mathbf{D}(\mathbf{u}), \quad \mathbf{D}(\mathbf{u}) = \frac{1}{2}\left(\mathbf{u}\nabla^T + \nabla\mathbf{u}^T\right). \tag{2}$$

For the boundary $\partial\Omega(t)$, $\mathbf{n}$ is the outward normal, $\boldsymbol{\alpha} = \alpha_{\|}\mathbb{I} + \left(\alpha_{\perp} - \alpha_{\|}\right)\mathbf{n}\mathbf{n}^T$ fixes the velocity at the boundary, $\boldsymbol{\beta} = \beta_{\|}\mathbb{I} + (\beta_{\perp} - \beta_{\|})\mathbf{n}\mathbf{n}^T$ fixes the traction at the boundary, $\mathbf{w}$ is the boundary movement velocity. Note, if both α_* and β_* are nonzero, then in SI units α_*/β_* is measured in $(\mathrm{m\,s}^{-1})$. Spaces H^1 and L^2 of problem unknowns are to be augmented at the boundary.

The equations of system (1) in domain $\Omega(t)$ in terms of a four-gradient read as:

$$\begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \begin{pmatrix} \nabla \\ \partial_t \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}. \tag{3}$$

Applying the divergence theorem to the four-dimensional integral of the left-hand side of (3) at cell $\omega(t)$ from the set of four-dimensional mesh cells $\mathcal{V}(\Omega(t))$ yields:

$$\int_{\omega(t)} \begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \begin{pmatrix} \nabla \\ \partial_t \end{pmatrix} \mathrm{d}V(t) = \sum_{\sigma(t) \in \mathcal{F}(\omega(t))} \int_{\sigma(t)} \begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \mathrm{d}\mathbf{S}(t), \tag{4}$$

where $\mathcal{F}(\omega(t))$ is a set of faces of a cell $\omega(t)$. Using the second-order approximation to the integral, we get

$$\sum_{\sigma(t) \in \mathcal{F}(\omega(t))} \int_{\sigma(t)} \begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \mathrm{d}\mathbf{S}(t) \approx \sum_{\sigma(t) \in \mathcal{F}(\omega(t))} |\sigma(t)| \begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{n} \\ n_t \end{pmatrix} \Bigg|_{\mathbf{x}_{\sigma(t)}}, \tag{5}$$

where $[\mathbf{n}\ n_t]^T$ is the four-dimensional normal to $\sigma(t)$, oriented outward of $\omega(t)$, $\mathbf{x}_{\sigma(t)}$ is the four-dimensional center, and $|\sigma(t)|$ is the area of face $\sigma(t)$.

The cornerstone of the finite-volume method is an approximation of the expression under the sum in (5):

$$\mathbf{F}|_{\mathbf{x}_{\sigma(t)}} := \begin{pmatrix} \mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u}) + p\mathbb{I} & \mathbf{u} \\ \mathbf{u}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{n} \\ n_t \end{pmatrix} \Bigg|_{\mathbf{x}_{\sigma(t)}} = \begin{pmatrix} \mathbf{u}n_t + \mathbf{u}\mathbf{u}^T\mathbf{n} - \boldsymbol{t}(\mathbf{u})\mathbf{n} + p\mathbf{n} \\ \mathbf{n}^T\mathbf{u} \end{pmatrix} \Bigg|_{\mathbf{x}_{\sigma(t)}}, \tag{6}$$

which corresponds to the coupled momentum and mass flux.

3.1 | Single-sided flux

Every cell $\omega(t) \in \mathcal{V}(\Omega(t))$ represents a moving finite volume. Let pressure p_i, velocity vector $\mathbf{u}_i$ and the four-gradient $\mathbf{G}_i$ of pressure and velocity be collocated at the four-dimensional center $\mathbf{x}_{\omega_i(t)}$ $\omega_i(t)$:

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{u}_i \\ p_i \end{bmatrix} \otimes \begin{bmatrix} \nabla \\ \partial_t \end{bmatrix} = \left(\begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} \otimes \begin{bmatrix} \nabla \\ \partial_t \end{bmatrix} \right) \Bigg|_{\mathbf{x}_{\omega_1(t)}}. \tag{7}$$

Here and after $A \bigotimes B$ is the Kronecker product. Pressure and velocity fields are reconstructed by piecewise-linear continuous functions, whereas their gradient is represented by piecewise-constant vector functions. We consider the approximation of separate parts of the coupled flux $\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$ from the side of the adjacent cell $\omega_1(t)$.

The second-order Taylor series in the proximity of $\mathbf{x}_{\omega_1(t)}$ for the inertia is

$$\mathbf{u}\mathbf{u}^T\mathbf{n}\big|_{\mathbf{x}_{\sigma(t)}} \approx \mathbf{u}\mathbf{u}^T\mathbf{n}\big|_{\mathbf{x}_{\omega_1(t)}} + \frac{\partial \mathbf{u}\mathbf{u}^T\mathbf{n}}{\partial \mathbf{u}^T}\bigg|_{\mathbf{x}_{\omega_1(t)}} \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \left(\mathbf{u} \otimes \begin{bmatrix} \nabla \\ \partial_t \end{bmatrix} \right) \Bigg|_{\mathbf{x}_{\omega_1(t)}}, \tag{8}$$

where

$$\frac{\partial \mathbf{u}\mathbf{u}^T\mathbf{n}}{\partial \mathbf{u}^T} = \mathbf{n}^T\mathbf{u}\mathbb{I} + \mathbf{u}\mathbf{n}^T, \quad \mathbf{u}\mathbf{u}^T\mathbf{n} = \frac{1}{2}\left(\mathbf{n}^T\mathbf{u}\mathbb{I} + \mathbf{u}\mathbf{n}^T\right)\mathbf{u}. \tag{9}$$

Let us introduce the unknown velocity vector $\mathbf{u}_\sigma$ at the center of the face $\mathbf{x}_{\sigma(t)}$. Using the assumption of linearity of the velocity field from (8) and (9) we obtain $\mathbf{u}\mathbf{u}^T\mathbf{n}|_{\mathbf{x}_{\sigma(t)}} \approx \frac{1}{2}(\mathbf{n}^T\mathbf{u}_1\mathbb{I} + \mathbf{u}_1\mathbf{n}^T)(2\mathbf{u}_\sigma - \mathbf{u}_1)$, and the first part of $\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$ reads as:

$$\begin{pmatrix} \mathbf{u}\mathbf{u}^T\mathbf{n} \\ \\ \end{pmatrix} \Bigg|_{\mathbf{x}_\sigma} \approx Q(\mathbf{u}_1, \mathbf{n}) \left(2\begin{bmatrix} \mathbf{u}_\sigma \\ p_\sigma \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix} \right), \quad Q(\mathbf{u}_1, \mathbf{n}) = \frac{1}{2}\begin{bmatrix} \mathbf{n}^T\mathbf{u}_1\mathbb{I} + \mathbf{u}_1\mathbf{n}^T & \\ & 0 \end{bmatrix}. \tag{10}$$

Due to the linearity of the velocity field, the traction term is approximated by:

$$-\boldsymbol{\tau}(\mathbf{u})\mathbf{n}|_{\mathbf{x}_{\sigma(t)}} = -\nu\left(\mathbf{u}\nabla^T + \nabla\mathbf{u}^T\right)\mathbf{n}\big|_{\mathbf{x}_{\sigma(t)}} \approx -\nu\left(\mathbb{I} \otimes \mathbf{n}^T + \mathbf{n}^T \otimes \mathbb{I}\right)(\mathbf{u} \otimes \nabla)|_{\mathbf{x}_{\omega_1(t)}}, \tag{11}$$

and in terms of the four-gradient it is expressed by:

$$\begin{pmatrix} -\boldsymbol{\tau}(\mathbf{u})\mathbf{n} \\ \\ \end{pmatrix} \Bigg|_{\mathbf{x}_{\sigma(t)}} \approx -W(\mathbf{n})\mathbf{G}_1, \quad W(\mathbf{n}) = \begin{bmatrix} \nu\mathbb{I} & \\ & 0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{n}^T & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{n}^T & 0 \end{bmatrix} \otimes \begin{bmatrix} \nu\mathbb{I} & \\ & 0 \end{bmatrix}. \tag{12}$$

Using the assumption of the linearity of velocity and pressure in $\omega_1(t)$ we decompose the gradient:

$$\mathbf{G}_1 = r_1^{-1}\mathbb{I} \otimes \mathbf{n}_t \left(\begin{bmatrix} \mathbf{u}_\sigma \\ p_\sigma \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix} \right) + \left(\mathbb{I} - r_1^{-1}\mathbb{I} \otimes \mathbf{n}_t \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \right)\mathbf{G}_1, \quad \mathbf{n}_t = \begin{bmatrix} \mathbf{n} \\ n_t \end{bmatrix}, \quad r_1 = \mathbf{n}_t \cdot \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right). \tag{13}$$

Using (13) in (12) we get the second part of $\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$:

$$\begin{pmatrix}-\boldsymbol{\tau}(\mathbf{u})\mathbf{n}\\ \\ \end{pmatrix}\Bigg|_{\mathbf{x}_{\sigma(t)}} \approx T\left(\mathbf{n}_t,\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)\left(\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix}-\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}\right)+\left(T\left(\mathbf{n}_t,\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)\otimes\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T-W(\mathbf{n})\right)\mathbf{G}_1, \tag{14}$$

where

$$T(\mathbf{n}_t,\mathbf{v})=(\mathbf{n}_t\cdot\mathbf{v})^{-1}W(\mathbf{n})(\mathbb{I}\otimes\mathbf{n}_t)=\frac{\nu}{\mathbf{n}_t\cdot\mathbf{v}}\begin{bmatrix}\mathbb{I}+\mathbf{n}\mathbf{n}^T & \\ & 0\end{bmatrix}. \tag{15}$$

Approximating the remaining part of the flux $\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$ we get:

$$\begin{pmatrix}\mathbf{u}n_t+p\mathbf{n}\\ \mathbf{u}^T\mathbf{n}\end{pmatrix}\Bigg|_{\mathbf{x}_{\sigma(t)}}=\begin{bmatrix}n_t\mathbb{I} & \mathbf{n}\\ \mathbf{n}^T & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix}\Bigg|_{\mathbf{x}_{\sigma(t)}}=D(\mathbf{n}_t)\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix},\quad D(\mathbf{n}_t)=\begin{bmatrix}n_t\mathbb{I} & \mathbf{n}\\ \mathbf{n}^T & 0\end{bmatrix} \tag{16}$$

Note that matrix coefficient $D(\mathbf{n}_t)$ in (16) is symmetric and indefinite with a negative eigenvalue $\frac{1}{2}\left(n_t-\sqrt{n_t^2+4}\right)$. It reflects the saddle-point nature of the system and leads to inf-sup stability issues. To overcome the issue, we introduce an additional stabilization term, vanishing for the linear velocity and pressure:

$$S(\mathbf{n})\left(\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix}-\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}\right)+S(\mathbf{n})\otimes\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\mathbf{G}_1=\mathbf{0},\quad S(\mathbf{n})=\begin{bmatrix}a_1(\mathbb{I}+\mathbf{n}\mathbf{n}^T) & \\ & b_1\end{bmatrix}, \tag{17}$$

where a_1 and b_1 are determined so that the eigenvalues of the matrix (19) are positive. In SI units, coefficient a_1 is measured by $(\mathrm{m\,s^{-1}})$, and b_1 by $(\mathrm{m^{-1}\,s})$.

Collecting (10), (14), (16), (17) and introducing $Q_1=Q(\mathbf{u}_1,\mathbf{n})$, $W_1=W(\mathbf{n})$, $T_1=T\left(\mathbf{n}_t,\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)$, $D_1=D(\mathbf{n}_t)$, $S_1=D(\mathbf{n}_t)+S(\mathbf{n})$, $\Lambda_1=T_1+S_1-D_1-2Q_1$ we obtain a semi-discrete coupled flux expression:

$$\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}\approx(T_1+S_1-Q_1)\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix}-\Lambda_1\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}+\left((T_1+S_1)\otimes\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T-W_1\right)\mathbf{G}_1. \tag{18}$$

Consider matrix coefficient Λ_1 for face unknowns in (18):

$$\Lambda_1=\begin{bmatrix}(a_1+r_1^{-1}\nu)(\mathbb{I}+\mathbf{n}\mathbf{n}^T)-(n_t\mathbb{I}+\mathbf{n}^T\mathbf{u})\mathbb{I}-\mathbf{u}\mathbf{n}^T & -\mathbf{n}\\ -\mathbf{n}^T & b_1\end{bmatrix}. \tag{19}$$

Maxima package[101] provides the eigenvalues of the Schur complement to the upper-left block:

$$\lambda_1=2a_1+2r_1^{-1}\nu-2\mathbf{n}^T\mathbf{u}_1-n_t-b_1^{-1},\quad \lambda_{2,3}=a_1+r_1^{-1}\nu-\mathbf{n}^T\mathbf{u}_1-n_t. \tag{20}$$

The positivity of eigenvalues in (20) is guaranteed by the following choice:

$$a_1=\max\left(|\mathbf{n}^T\mathbf{u}_1|+|\mathbf{n}^T\mathbf{u}_1+n_t|-r_1^{-1}\nu,\epsilon\right),\quad b_1=\left(a_1+r_1^{-1}\nu+\left|\mathbf{n}^T\mathbf{u}_1\right|+\left|\mathbf{n}^T\mathbf{u}_1+n_t\right|-\left(2\mathbf{n}^T\mathbf{u}_1+n_t\right)\right)^{-1}, \tag{21}$$

where $\epsilon=10^{-5}$ is a small positive value. According to Haynsworth inertia additivity formula,[102] eigenvalues of Λ_1 are all positive.

3.2 | Internal flux

Consider an internal face $\sigma(t) \in \mathcal{F}(\Omega(t))$ adjacent to cells $\omega_1(t)$ and $\omega_2(t)$, $\sigma(t) = \omega_1(t) \cap \omega_2(t)$, $\omega_1(t), \omega_2(t) \in \mathcal{V}(\Omega(t_{n+1}))$. We assume that four-dimensional normal $\mathbf{n}_t$ to face $\sigma(t)$ is oriented outside of $\omega_1(t)$ and inside $\omega_2(t)$.

Denote $Q_2 = Q(\mathbf{u}_2, -\mathbf{n})$, $W_2 = W(-\mathbf{n})$, $T_2 = T\left(-\mathbf{n}_t, \mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)$, $D_2 = D(-\mathbf{n}_t)$, $S_2 = S(-\mathbf{n})$, $\Lambda_2 = T_2 + S_2 - D_2 - 2Q_2$. Using pressure, velocity, and the four-gradient at the center of the cell $\omega_2(t)$ and unknown pressure and velocity at the face center $\sigma(t)$, we get an approximation of the coupled flux $\mathbf{F}$ at the center of the face $\sigma(t)$ from the side of the cell $\omega_2(t)$:

$$\mathbf{F}|_{\mathbf{x}_{\sigma(t)}} \approx \Lambda_2 \begin{bmatrix} \mathbf{u}_\sigma \\ p_\sigma \end{bmatrix} - (T_2 + S_2 - Q_2) \begin{bmatrix} \mathbf{u}_2 \\ p_2 \end{bmatrix} - \left((T_2 + S_2) \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T - W_2\right) \mathbf{G}_2. \tag{22}$$

Equating the approximations (18) and (22) we obtain unknown pressure and velocity at face center:

$$\begin{bmatrix} \mathbf{u}_\sigma \\ p_\sigma \end{bmatrix} = (\Lambda_1 + \Lambda_2)^{-1} \begin{pmatrix} (T_1 + S_1 - Q_1) & \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix} + \left((T_1 + S_1) \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T - W_1\right) \mathbf{G}_1 \\ +(T_2 + S_2 - Q_2) & \begin{bmatrix} \mathbf{u}_2 \\ p_2 \end{bmatrix} + \left((T_2 + S_2) \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T - W_2\right) \mathbf{G}_2 \end{pmatrix}. \tag{23}$$

Using (23) in either (18) or (22) we obtain the unique coupled flux approximation:

$$\begin{aligned} \mathbf{F}|_{\mathbf{x}_{\sigma(t)}} &\approx \Lambda_2 (\Lambda_1 + \Lambda_2)^{-1} \left((T_1 + S_1 - Q_1) \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix} + \left((T_1 + S_1) \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T - W_1\right) \mathbf{G}_1 \right) \\ &\quad - \Lambda_1 (\Lambda_1 + \Lambda_2)^{-1} \left((T_2 + S_2 - Q_2) \begin{bmatrix} \mathbf{u}_2 \\ p_2 \end{bmatrix} + \left((T_2 + S_2) \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T - W_2\right) \mathbf{G}_2 \right). \end{aligned} \tag{24}$$

3.3 | Boundary flux

Consider a boundary face $\sigma(t) \in \mathcal{F}(\partial\Omega(t))$ adjacent to cell $\omega_1(t) \in \mathcal{V}(\Omega(t_{n+1}))$, $\sigma(t) = \partial\Omega(t) \cap \omega_1(t)$.

Following Reference [25], we introduce an additional condition from the normal-projected momentum conservation equation:

$$\mathbf{n} \cdot \nabla p|_{\mathbf{x}_{\sigma(t)}} = \left(\mathbf{n} \cdot \mathbf{f} - \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial t} - \mathbf{n} \cdot \operatorname{div}\left(\mathbf{u}\mathbf{u}^T - \boldsymbol{\tau}(\mathbf{u})\right) \right) \Big|_{\mathbf{x}_{\sigma(t)}}. \tag{25}$$

Considering the assumption of linearity of pressure and velocity fields, we omit the traction term. The constant pressure gradient allows us to consider (25) at the center of cell $\mathbf{x}_{\omega_1(t)}$ instead of $\mathbf{x}_{\sigma(t)}$. Thus we obtain the following approximation:

$$\mathbf{n} \cdot \nabla p|_{\mathbf{x}_{\omega_1(t)}} = \mathbf{n} \cdot \mathbf{f}_1 - \begin{bmatrix} \mathbf{n}^T & 0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{u}_1^T & 1 \end{bmatrix} \mathbf{G}_1 \tag{26}$$

where $\mathbf{f}_1 = \mathbf{f}|_{\mathbf{x}_{\omega_1(t)}}$.

We combine the boundary condition equations from (1) with (26) to get

$$B_D\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}+B_N W_b \mathbf{G}_1 = R_b,\quad B_D=\begin{bmatrix}\boldsymbol{\alpha} & -\boldsymbol{\beta}\mathbf{n}\\ & \end{bmatrix},\quad B_N=\begin{bmatrix}\boldsymbol{\beta} & \\ & 1\end{bmatrix},\quad R_b=\begin{bmatrix}\mathbf{r}+\boldsymbol{\alpha}\mathbf{w}\\ \mathbf{n}\cdot\mathbf{f}_1\end{bmatrix} \tag{27}$$

with

$$W_b=\begin{bmatrix}\nu\mathbb{I} & \\ & 1\end{bmatrix}\otimes\begin{bmatrix}\mathbf{n}^T & 0\end{bmatrix}+\begin{bmatrix}\mathbf{n}^T & 0\end{bmatrix}\otimes\begin{bmatrix}\nu\mathbb{I} & \\ \mathbf{u}_1^T & 1\end{bmatrix}. \tag{28}$$

We use the gradient decomposition (13) and add (17) in (27) to get:

$$\Lambda_b\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}+\left(B_N W_b-(B_N T_b+S_b)\otimes\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\right)\mathbf{G}_1=R_b+(B_N T_b+S_b)\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix}, \tag{29}$$

where $\Lambda_b=B_D+B_N T_b+S_b$, T_b and S_b are given by:

$$T_b=r_1^{-1}W_b(\mathbb{I}\otimes\mathbf{n}_t)=r_1^{-1}\begin{bmatrix}\nu(\mathbb{I}+\mathbf{n}\mathbf{n}^T) & \\ (\mathbf{n}^T\mathbf{u}_1+n_t)\mathbf{n}^T & 1\end{bmatrix},\quad S_b=\begin{bmatrix}a_b\boldsymbol{\beta}\mathbf{n}\mathbf{n}^T & \\ & 0\end{bmatrix}. \tag{30}$$

The stabilization parameter a_b, is chosen to make the eigenvalues of Λ_b positive:

$$\Lambda_b=\begin{bmatrix}\boldsymbol{\alpha}+r_1^{-1}\nu\boldsymbol{\beta}(\mathbb{I}+\mathbf{n}\mathbf{n}^T)+a_b\boldsymbol{\beta}\mathbf{n}\mathbf{n}^T & -\boldsymbol{\beta}\mathbf{n}\\ r_1^{-1}(\mathbf{n}^T\mathbf{u}_1+n_t)\mathbf{n}^T & r_1^{-1}\end{bmatrix}. \tag{31}$$

To this end, we consider the eigenvalues of the Schur complement to the upper-left block, which are:

$$\lambda_1=\alpha_\perp+\beta_\perp\left(a_b+2r_1^{-1}\nu+\mathbf{n}^T\mathbf{u}_1+n_t\right),\quad \lambda_{2,3}=\alpha_\parallel+r_1^{-1}\nu\beta_\parallel. \tag{32}$$

Requiring positivity of λ_1 for nonzero $\beta_\perp$ results in the following expression:

$$a_b=\max\left(-\alpha_\perp/\beta_\perp-2r_1^{-1}\nu-\mathbf{n}^T\mathbf{u}_1-n_t,\epsilon\right), \tag{33}$$

where $\epsilon=10^{-5}$ is a small positive value.

Solving (29) for unknowns at face $\sigma(t)$ we get

$$\begin{bmatrix}\mathbf{u}_\sigma\\ p_\sigma\end{bmatrix}=\Lambda_b^{-1}\left(R_b+(B_N T_b+S_b)\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix}+\left((B_N T_b+S_b)\otimes\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T-B_N W_b\right)\mathbf{G}_1\right). \tag{34}$$

Finally, using (34) in (18) we obtain the boundary flux $\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$.

3.4 | Temporal boundary flux

The coupled fluxes at the top temporal boundary $\overline{\sigma}(t)=\omega_1(t_{n+1})\cap\omega_1(t_{n+2})$ and the bottom temporal boundary $\underline{\sigma}(t)=\omega_1(t_n)\cap\omega_1(t_{n+1})$ are:

$$\mathbf{F}|_{\overline{\sigma}(t)} = \begin{bmatrix} \mathbf{u}_1^{n+1} \\ 0 \end{bmatrix}, \quad \mathbf{F}|_{\underline{\sigma}(t)} = -\begin{bmatrix} \mathbf{u}_1^{n} \\ 0 \end{bmatrix}, \tag{35}$$

with the four-dimensional areas corresponding to volumes $\omega_1(t_{n+1})$ and $\omega_1(t_n)$, respectively. In the case of a non-deforming mesh, (35) results in the backward Euler approximation of $\partial\mathbf{u}/\partial t$.

3.5 | Gradient reconstruction

We reconstruct the four-gradient $\mathbf{G}_1$ at the center of each time level $\omega_i(t) \in \Omega(t_{n+1})$. For a cell $\omega_1(t)$ and every other adjacent cell $\omega_2(t) \in \mathcal{V}(\Omega(t_{n+1}))$, $\omega_1 \cap \omega_2 \neq \emptyset$, $\omega_2 \neq \omega_1$, we consider the following condition for the gradient:

$$\mathbb{I} \otimes \left(\mathbf{x}_{\omega_2(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \mathbf{G}_1 = \begin{bmatrix} \mathbf{u}_2 \\ p_2 \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix}. \tag{36}$$

At the boundary with prescribed conditions $\sigma(t) \in \mathcal{F}(\partial\Omega(t_{n+1}))$, $\sigma(t) = \partial\Omega(t) \cap \omega_1(t)$, $\omega_1(t) \in \mathcal{V}(\Omega(t_{n+1}))$, we derive the condition from (34):

$$\left(B_D \otimes \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T + B_N W_b\right) \mathbf{G}_1 = R - B_D \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix}. \tag{37}$$

An additional condition over the temporal boundary is added:

$$\mathbb{I} \otimes \left(\mathbf{x}_{\omega_1(t_n)} - \mathbf{x}_{\omega_1(t_{n+1})}\right)^T \mathbf{G}_1 = \begin{bmatrix} \mathbf{u}_1^n \\ p_1^n \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ p_1 \end{bmatrix}. \tag{38}$$

As a result, in d-dimensions every condition (36) and (37) provides $d+1$ conditions for the four-gradient $\mathbf{G}_1$, consisting of $(d+1)^2$ unknowns. It is sufficient to consider $(d+1)$ conditions. Gathering conditions for a cell $\omega_1(t)$ we obtain a system $A\mathbf{G}_1 = b$, solved with the Cholesky method: $\mathbf{G}_1 = \left(A^T A\right)^{-1} A^T b$. Note that the contribution of (37) into A is nonlinear due to the dependence of W_b on $\mathbf{u}_1$. We also scale the last row of (37) by $r_1^{3/2}$.

The condition $\omega_2(t) \in \mathcal{V}(\Omega(t_{n+1}))$, $\omega_1 \cap \omega_2 \neq \emptyset, \omega_2 \neq \omega_1$ includes every cell sharing at least a node with the cell ω_1 and results in rather wide gradient stencil. We require such stencil for the method stability on a moving mesh that results in complex geometric configurations. It may significantly degrade the performance of the method on regular meshes and shall be addressed in future.

4 | GEOMETRY IN FOUR DIMENSIONS

The domain geometry Ω_n in temporal layer n is assumed to be known, and each layer n corresponds to time t_n all over Ω_n, as illustrated in Figure 1.

Three-dimensional cells $\omega^{n+1} = \omega(t_{n+1})$ possess collocation points with three-dimensional coordinates $\mathbf{x}_{\omega^{n+1}}$ and the fourth coordinate t_{n+1}.

Evolution of a four-dimensional cell $\omega_i(t_n, t_{n+1})$ from time level t_n to time level t_{n+1} is represented by two three-dimensional cells ω_i^n and ω_i^{n+1}. Thus, it is a prism with two bases and the volume computed by

$$|\,\omega_i(t_n, t_{n+1})\,| = \left(|\omega_i^n| + |\omega_i^{n+1}|\right)(t_{n+1} - t_n)/2. \tag{39}$$

The volume is measured in SI by (m^3 s).

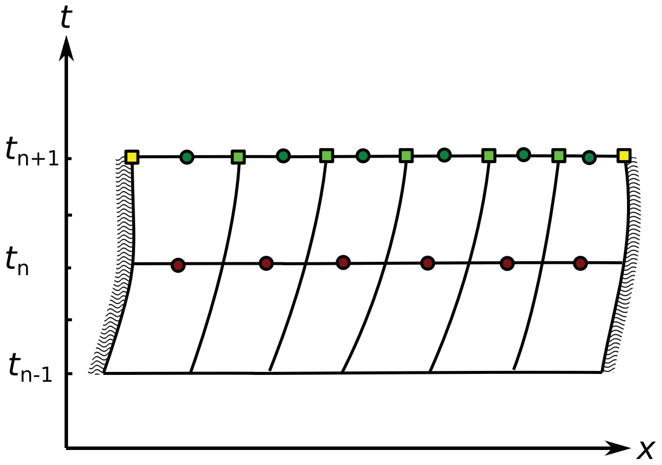

FIGURE 1 Representation of $\Omega(t)$ for a single-dimensional problem. Green circles are collocation points at t_{n+1}, red circles are collocation points at t_n. Green squares are points with approximations of fluxes at internal faces, yellow squares—at boundary faces. Temporal boundary fluxes are approximated at green and red circles.

Similarly, the four-dimensional face $\sigma_i(t_n,t_{n+1})$ between time moments t_n and t_{n+1} is framed between two three-dimensional faces σ_i^n and σ_i^{n+1} with the area computed by

$$|\sigma_i(t_n,t_{n+1})| = \left(|\sigma_i^n| + |\sigma_i^{n+1}|\right)(t_{n+1}-t_n)/2. \tag{40}$$

The area is measured in SI by (m^2 s).

Let $\mathbf{x}_{\sigma_i^n}, \mathbf{x}_{\sigma_i^{n+1}}$ correspond to three-dimensional centers, and $\mathbf{n}_{\sigma_i^n}, \mathbf{n}_{\sigma_i^{n+1}}$ correspond to three-dimensional normal to faces $\sigma_i^n, \sigma_i^{n+1}$, respectively. We define a three-dimensional normal $\mathbf{n} = \left(\mathbf{n}_{\sigma_i^n} + \mathbf{n}_{\sigma_i^{n+1}}\right)/2$ at the midpoint between layers n and $(n+1)$ and introduce the fourth coordinate of normal, satisfying

$$\begin{bmatrix}\mathbf{n}^T & n_t\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\sigma_i^n} \\ t_n\end{bmatrix} = \begin{bmatrix}\mathbf{n}^T & n_t\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\sigma_i^{n+1}} \\ t_{n+1}\end{bmatrix}, \tag{41}$$

thus determining a plane passing through the four-dimensional centers of $\sigma_i^n, \sigma_i^{n+1}$.

We obtain

$$n_t = -\mathbf{n}^T\left(\mathbf{x}_{\sigma_i^{n+1}} - \mathbf{x}_{\sigma_i^n}\right)/(t_{n+1}-t_n) \approx -\mathbf{n}^T\mathbf{w}_{\sigma(t_n,t_{n+1})}, \tag{42}$$

where $\mathbf{w}_{\sigma(t_n,t_{n+1})}$ is a mesh movement velocity. Note that the SI units of n_t correspond to velocity (m s^{-1}). Since the four-dimensional coordinate is measured in $[m\ m\ m\ s]^T$, and the normal is measured in (111 m s^{-1}), the scalar product between a four-dimensional coordinate and a four-dimensional normal is measured in (m). The divergent formula for the volume $|\omega| = \sum_\sigma |\sigma| \mathbf{n}^T\mathbf{x}$ yields (m^3 s).

The outward four-dimensional normal to the temporal layer t_{n+1} is $[0\ 0\ 0\ 1]$, the area of a four-dimensional face corresponds to the volume of the three-dimensional cell $|\omega_i^{n+1}|$. By analogy, the normal to temporal layer t_n is $[0\ 0\ 0\ -1]$, and the area corresponds to the volume $|\omega_i^n|$. The units of normal are $[1\ 1\ 1\ s]$.

In numerical experiments, we use (mm) units for the length.

5 | PROBLEM SOLUTION

We use the Newton method to obtain the solution at time level t_{n+1}. The initial solution is taken from the previous time step $\mathbf{u}^{n+1,0} = \mathbf{u}^n$ and $p^{n+1,0} = p^n$. At the kth Newton iteration, we assemble the residual $\mathscr{R}^k$ and corresponding Jacobian J^k:

$$\mathcal{R}^k \in \mathfrak{R}^{4|\mathcal{V}(\Omega(t_{n+1})\times 1)}, \quad \mathcal{J}^k \in \mathfrak{R}^{4|\mathcal{V}(\Omega(t_{n+1}))|\times 4|\mathcal{V}(\Omega(t_{n+1}))|}. \tag{43}$$

The Jacobian is obtained using automatic differentiation; thus, only residual computation is of concern. Automatic differentiation significantly degrades the performance, but facilitates the research of various approaches. We compute the residual using the following steps:

1. Compute the gradient $\mathbf{G}_i$ with the derivatives at every cell $\omega_i(t) \in \mathcal{V}(\Omega(t_{n+1}))$ following Section 3.5.
2. Compute the residual at every cell $\omega_i(t) \in \mathcal{V}(\Omega(t_{n+1}))$ for divergent terms:

$$\mathcal{R}_i^k = \sum_{\sigma(t)} |\,\sigma(t)\,|\,\mathbf{F}|_{\mathbf{x}_{\sigma(t)}}, \tag{44}$$

where the coupled flux is computed using (24) and (18)–(34) depending on the face type. At the temporal boundary $\overline{\sigma}(t)$ and $\underline{\sigma}(t)$, add the coupled flux using (35).

3. For every cell $\omega_i(t) \in \mathcal{V}(\Omega(t_{n+1}))$, subtract from the velocity components of $\mathcal{R}_i^k$, the volumetric forces $\mathbf{f}_i = \mathbf{f}|_{\mathbf{x}_{\omega_i(t)}}$, multiplied by $|\,\omega_i(t_n, t_{n+1})\,|$

The convergence criterion for the Newton method is $\|\,\mathcal{R}^k\,\| \leq \max\left(\tau_{abs}, \tau_{rel}\,\|\,\mathcal{R}^0\,\|\right)$. If the convergence criteria are met, we set $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,k}$ and $p^{n+1} = p^{n+1,k}$ and proceed to the mesh adaptation, followed by the next time step. Otherwise, the pressure and velocity updates are determined from the solution of the linear system $\mathcal{J}^k\left[\Delta\mathbf{u}^k \;\; \Delta p^k\right]^T = -\mathcal{R}^k$. The $(k+1)$th pressure and velocity are determined by $\mathbf{u}^{n+1,k+1} = \mathbf{u}^{n+1,k} + \alpha\Delta\mathbf{u}^k$ and $p^{n+1,k+1} = p^{n+1,k} + \alpha\Delta p^k$. Heuristics are used to select the update parameter α at every Newton step. The first technique prevents stagnation in the Newton method. To this end, we count the number q of repetitions of the maximal residual error at the same position if $i = \arg\max_i\left(\mathcal{R}_i^m\right) = \arg\max_i\left(\mathcal{R}_i^k\right)$ and $|\,\mathcal{R}_i^k - \mathcal{R}_i^m\,| \leq 10^{-3}\,|\,\mathcal{R}_i^k\,|$ for every $m < i$. Then $\alpha = (3/4)^q$. The second technique reduces α to limit the Newton update not to exceed more than twice the mean of the maximum update over previous Newton steps.

If convergence is not attained after the prescribed number of nonlinear iterations, the time step is halved. Upon success, the time step is doubled if the maximum time step is not attained.

At every Newton iteration, the linear system for the update is solved iteratively with the multilevel preconditioner.[103,104] The iterative convergence tolerances are $\tau_{abs} = 10^{-12}$, $\tau_{rel} = 10^{-18}$, dropping tolerances in the second-order incomplete factorization are $\tau_1 = 10^{-3}$ and $\tau_2 = 10^{-5}$, pivoting by condition estimation is $\kappa = 2.5$. In parallel, a single overlapping layer is used for the additive Schwarz method. We shall note that the block-structured preconditioners such as point-block algebraic multigrid method[105] appear to be much more efficient for the problem in question.

Once the solution for the next time step is obtained, we perform coarsening of the three-dimensional mesh, followed by refinement of the three-dimensional mesh and mesh balancing. The adaptation steps are performed in sweeps until all the marked elements are changed and no new elements are marked for the adaptation.[80,91] During adaptation, data transfer is of concern. The adaptation operates with the data of the converged solution at the next time level, t^{n+1}. Thereby, both the previous step solution and geometry are no longer needed. For the refinement, we reuse gradients $\mathbf{G}_i$, computed at the last Newton iteration. During the refinement step, the interpolation from a coarse cell $\omega_1(t)$ to fine cells $\omega_i(t)$ is computed by

$$\begin{bmatrix}\mathbf{u}_i\\ p_i\end{bmatrix} = \begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix} + \Theta \otimes \left(\mathbf{x}_{\omega_i(t)} - \mathbf{x}_{\omega_1(t)} \;\; 0\right)^T \mathbf{G}_1, \quad \mathbf{G}_i = \mathbf{G}_1, \tag{45}$$

where $\Theta = diag\left(\theta_u, \theta_v, \theta_w, \theta_p\right)$ is chosen to limit the interpolation for all new cells ω_i:

$$\min_{\omega_j(t) \,\in\, \mathcal{V}_n(\omega_1(t))}\left(\begin{bmatrix}\mathbf{u}_j\\ p_j\end{bmatrix}\right) \leq \begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix} + \Theta \otimes \left(\mathbf{x}_{\omega_i(t)} - \mathbf{x}_{\omega_1(t)} \;\; 0\right)^T \mathbf{G}_1 \leq \max_{\omega_j(t) \,\in\, \mathcal{V}_n(\omega_1(t))}\left(\begin{bmatrix}\mathbf{u}_i\\ p_i\end{bmatrix}\right), \tag{46}$$

here $\mathcal{V}_n(\omega_1(t))$ is a set of cells in Ω_{n+1} sharing at least a node with the cell $\omega_1(t)$. In (46) each component can be considered separately. The interpolation is conservative under condition $\sum_i |\,\omega_i(t)\,|\,\mathbf{x}_i = |\,\omega_1(t)\,|\,\mathbf{x}_1$ and is monotone due to (46). During coarsening of fine cells $\omega_i(t)$ to a coarse cell $\omega_1(t)$ we use the simple averaging:

$$\begin{bmatrix}\mathbf{u}_1\\ p_1\end{bmatrix} = |\omega_1(t)|^{-1}\sum_i |\omega_i(t)|\begin{bmatrix}\mathbf{u}_i\\ p_i\end{bmatrix},\quad \mathbf{G}_1 = |\omega_1(t)|^{-1}\sum_i |\omega_i(t)|\mathbf{G}_i, \tag{47}$$

which is both monotone and conservative.

6 | VERIFICATION TESTS

For verification tests we use units for distance (mm), mass (g), time (s), velocity in (mm s^{-1}), blood density $\rho = 1060$ (kg m^{-3}) $= 1.06\cdot 10^{-3}$ (g mm^{-3}) and dynamic viscosity $\mu = 3.5$ [cP] $= 3.5\times 10^{-3}$ (g mm^{-3} s^{-1}). The kinematic viscosity is $\nu = \mu/\rho \approx 3.3$ (mm^2 s^{-1}). All analytical expressions are: averaged over mesh elements using the seventh-order integration formula. The nonlinear convergence criteria are $\tau_{\text{abs}} = 10^{-6}$, $\tau_{\text{rel}} = 10^{-5}$. We recall that for the solution of the Navier–Stokes equations we assign 4 unknowns to each computational cell.

6.1 | Ethier–Steinman analytical solution

We first consider the Ethier–Steinman analytical solution[106]

$$\mathbf{u}(x,y,z,t) = -\frac{\pi}{4}\exp\left(-\nu\frac{\pi^2 t}{4}\right)\begin{bmatrix} \exp\left(\frac{\pi x}{4}\right)\sin\left(\frac{\pi(y+2z)}{4}\right) + \exp\left(\frac{\pi z}{4}\right)\cos\left(\frac{\pi(x+2y)}{4}\right)\\ \exp\left(\frac{\pi y}{4}\right)\sin\left(\frac{\pi(z+2x)}{4}\right) + \exp\left(\frac{\pi x}{4}\right)\cos\left(\frac{\pi(y+2z)}{4}\right)\\ \exp\left(\frac{\pi z}{4}\right)\sin\left(\frac{\pi(x+2y)}{4}\right) + \exp\left(\frac{\pi y}{4}\right)\cos\left(\frac{\pi(z+2x)}{4}\right)\end{bmatrix}, \tag{48}$$

$$p(x,y,z,t) = -\frac{\pi^2}{32}\exp\left(-\nu\frac{\pi^2 t}{2}\right)\begin{pmatrix} \exp\left(\frac{\pi x}{2}\right) + \exp\left(\frac{\pi y}{2}\right) + \exp\left(\frac{\pi z}{2}\right)\\ +2\sin\left(\frac{\pi(x+2y)}{4}\right)\cos\left(\frac{\pi(z+2x)}{4}\right)\exp\left(\frac{\pi(y+z)}{4}\right)\\ +2\sin\left(\frac{\pi(y+2z)}{4}\right)\cos\left(\frac{\pi(x+2y)}{4}\right)\exp\left(\frac{\pi(z+x)}{4}\right)\\ +2\sin\left(\frac{\pi(z+2x)}{4}\right)\cos\left(\frac{\pi(y+2z)}{4}\right)\exp\left(\frac{\pi(x+y)}{4}\right)\end{pmatrix}. \tag{49}$$

We define the spherical domain $\Omega(t) = \{\mathbf{x} : \|\mathbf{x} - t\mathbf{w}\| \le 5\cdot 10^{-1}\ [mm]\}$ deforming with velocity $\mathbf{w}$ (mm s^{-1}). The Dirichlet boundary condition is isimposed on $\Gamma_D = \partial\Omega$ by setting $\alpha_\perp = \alpha_\| = 1$, $\beta_\perp = \beta_\| = 0$ and the right hand side by the analytical solution in (48).

The solution to the pressure is not unique, therefore, we add a constraint for the pressure integral:

$$\int_\Omega p(x,y,z,t_{n+1})\mathrm{d}V \approx \sum_{\omega_i\,\in\,\mathcal{V}(\Omega(t))} p(x_i,y_i,z_i,t_{n+1})|\omega_i| = \sum_{\omega_i\,\in\,\mathcal{V}(\Omega(t))} p_i|\omega_i|. \tag{50}$$

Convergence study for the steady and deforming domains on a sequence of quasi-uniform meshes $\Omega_1,\ldots,\Omega_4$ without adaptation are given in Table 1. It demonstrates the second-order convergence for the velocity and the first-order convergence for the pressure. The orders of convergence are similar for the steady and deforming domains. Table 2 presents error norms on meshes Ω_i^a obtained by refinement of Ω_i the layer of cells next to the boundary. The error reduces both for velocity and pressure. Therefore, the interface between coarse and fine cells does not generate spurious errors, which are often observed on locally refined meshes.[13] However, the order of convergence is slightly worse on the adapted meshes. The original mesh Ω_3 and the locally refined mesh Ω_3^a are displayed in Figure 2. Each time step requires three to four Newton iterations for the nonlinear problem's convergence.

TABLE 1 Error norms for the Ethier–Steinman problem on quasi-uniform meshes.

			Steady, w = 0 (mm s^{-1})		**Moving, w = 1 (mm s^{-1})**	
$\Omega(t)$	Δt	**#Cells**	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$
Ω_1	1/50	249	3.96×10^{-3}	2.16×10^{-1}	4.84×10^{-3}	2.07×10^{-1}
Ω_2	1/100	1343	1.22×10^{-3}	9.34×10^{-2}	1.49×10^{-3}	8.89×10^{-2}
Ω_3	1/200	9748	3.00×10^{-4}	3.76×10^{-2}	3.61×10^{-4}	3.49×10^{-2}
Ω_4	1/400	67,405	8.51×10^{-5}	1.67×10^{-2}	9.94×10^{-5}	1.53×10^{-2}
Rate			1.82	1.17	1.86	1.19

TABLE 2 Error norms for the Ethier–Steinman problem on locally refined meshes.

			Steady, w = 0 (mm s^{-1})		**Moving, w = 1 (mm s^{-1})**	
$\Omega(t)$	Δt	**#Cells**	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$
Ω_1^a	1/50	1866	1.36×10^{-3}	1.27×10^{-1}	1.64×10^{-3}	9.93×10^{-2}
Ω_2^a	1/100	8168	5.36×10^{-4}	6.78×10^{-2}	6.34×10^{-4}	5.97×10^{-2}
Ω_3^a	1/200	40,604	1.53×10^{-4}	3.09×10^{-2}	1.70×10^{-4}	2.73×10^{-2}
Ω_4	1/400	193,272	5.21×10^{-5}	1.50×10^{-2}	5.04×10^{-5}	1.34×10^{-2}
Rate			1.55	1.04	1.75	1.03

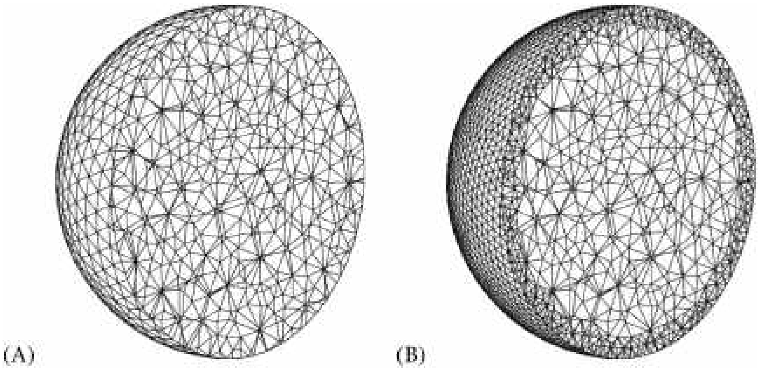

FIGURE 2 Middle cutaway of the tetrahedral mesh Ω_3 (A) and Ω_3^a (B) for the sphere.

6.2 | Analytical solution in a shrinking cylinder

Following Reference [99,107], we consider the flow in the shrinking cylindrical domain $\Omega(t)$:

$$\Omega(t)=\left\{\mathbf{x}=(x,y,z):-4\ (\mathrm{mm})\le z\le 4\ (\mathrm{mm}),\ \ x^2+y^2\le\exp\left(\frac{z}{4}+1\right)\left(1-\frac{1}{4}t\right)\ (\mathrm{mm}^2)\right\},\quad t\in[0,0.2]\ [s]. \tag{51}$$

The analytical solution $\mathbf{u}$, p, and corresponding mesh movement velocity $\mathbf{w}$ read as

$$\begin{aligned}\mathbf{u}(x,y,z,t)\ &=\frac{8}{4-t}\begin{bmatrix}0\\0\\1\end{bmatrix}-2r^2\frac{\exp\left(-\frac{z}{4}-1\right)}{(4-t)^2}\begin{bmatrix}x\\y\\16\end{bmatrix},\quad \mathbf{w}(x,y,z,t)=\frac{1}{2(4-t)}\begin{bmatrix}-x\\-y\\0\end{bmatrix},\\ p(x,y,z,t)\ &=\frac{8}{(4-t)^2}\left(4-z+64\nu\left(\exp\left(-\frac{z}{4}-1\right)-\exp(-2)\right),\right)\end{aligned} \tag{52}$$

where $r=\sqrt{x^2+y^2}$.

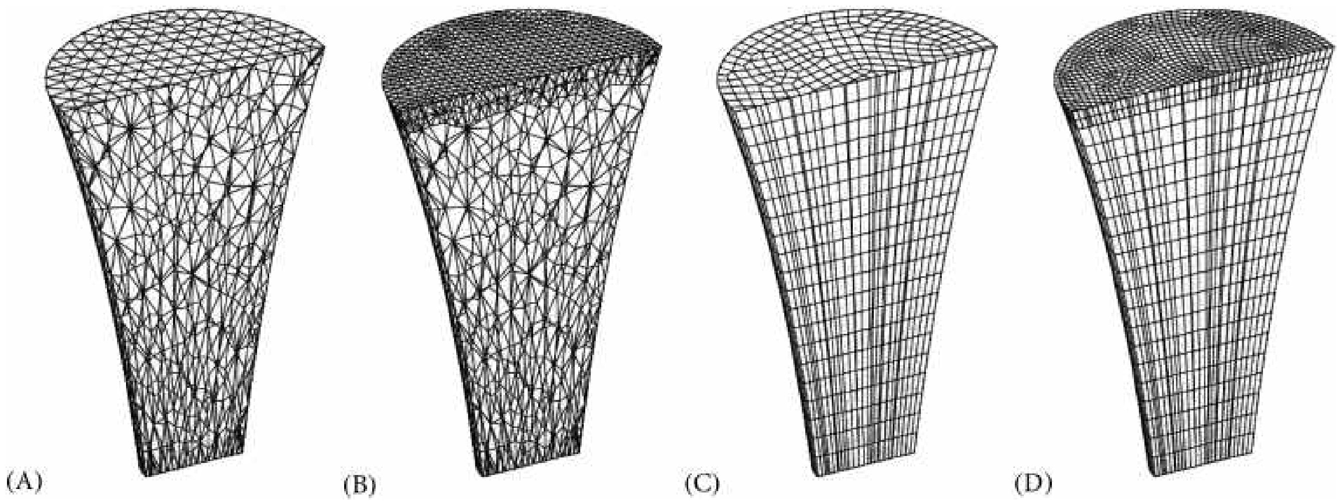

FIGURE 3 Middle cutaway of the deformed cylinder mesh at the initial state, tetrahedral mesh Ω_3 (A), locally refined tetrahedral mesh Ω_3^a (B), hexahedral mesh Ω_3 (C), and locally refined hexahedral mesh Ω_3^a (D).

The solution (52) produces in (1) the right-hand side

$$\mathbf{f} = \nu \frac{\exp\left(-\frac{z}{4}-1\right)}{8(4-t)^2}\begin{bmatrix}128x+xr^2\\128y+yr^2\\16r^2\end{bmatrix} - 4r^4\frac{\exp\left(-\frac{z}{2}-2\right)}{(4-t)^4}\begin{bmatrix}x\\y\\32\end{bmatrix}. \tag{53}$$

At the top boundary $\Gamma_t = \partial\Omega|_{z=4}$ a non-homogeneous Neumann condition is set by $\alpha_\perp = \alpha_\parallel = 0$ and $\beta_\perp = \beta_\parallel = 1$, whereas at the bottom boundary $\Gamma_b = \partial\Omega|_{z=-4}$ a non-homogeneous Dirichlet condition Γ_D is set by $\alpha_\perp = \alpha_\parallel = 1$ and $\beta_\perp = \beta_\parallel = 0$. At the remaining boundary $\Gamma_s = \partial\Omega\backslash(\Gamma_t \cap \Gamma_b)$, the homogeneous Dirichlet condition is set. The right hand side $\mathbf{r}$ at each boundary part is prescribed according to

$$(\boldsymbol{\tau}(\mathbf{u}) - p)\mathbf{n}|_{\Gamma_t} = \nu\frac{\exp(-2)}{2(4-t)^2}\begin{bmatrix}-128x+xr^2\\-128y+yr^2\\32r^2\end{bmatrix}, \quad (\mathbf{u}-\mathbf{w})|_{\Gamma_b} = \frac{4-t-4r^2}{2(4-t)^2}\begin{bmatrix}x\\y\\16\end{bmatrix}, \quad (\mathbf{u}-\mathbf{w})|_{\Gamma_s} = \mathbf{0}. \tag{54}$$

We solve (1) on a sequence of tetrahedral and hexahedral meshes, see Figure 3A,C. Convergence study for these cases is presented in Table 3. The convergence rates for both velocity and pressure are of the first-order accuracy. We refine the mesh cells along the boundary Γ_t with the outflow condition, rf. Figure 3B,D, and present convergence study in Table 4. Refinement along the outflow boundary improves the errors on both types of locally refined meshes. The solution of the nonlinear system at each time step requires two to five nonlinear iterations.

The above numerical results demonstrate that the proposed method is able to solve the Navier–Stokes equations in deforming domains for the blood kinematic viscosity. Moreover, the method improves with mesh refinement and does not generate artificial errors along the refinement interface.

7 | BLOOD FLOW IN RIGHT VENTRICLE

We solve the Navier–Stokes equations in a domain segmented from a time series of computed tomography scans. The domain corresponds to the right ventricle of a patient with transposition of the great arteries, a rare congenital defect. The domain evolution is provided by 90 topologically invariant meshes[99] with 13,222 nodes, 86,920 edges, and 70,533 tetrahedra, see Figure 4. For details of mesh generation process we refer to Reference [108]. The domain was segmented

TABLE 3 Error norms for the shrinking cylinder problem on non-refined meshes.

		Tetrahedral			Hexahedral		
$\Omega(t)$	Δt	#Cells	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$	#cells	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$
Ω_1	1/25	376	4.95×10^{-2}	5.05×10^{-1}	160	1.04×10^{-1}	1.52×10^{-0}
Ω_2	1/50	1668	1.33×10^{-2}	5.40×10^{-1}	1060	1.56×10^{-2}	3.32×10^{-1}
Ω_3	1/100	10,186	4.77×10^{-3}	2.43×10^{-1}	7740	3.65×10^{-3}	1.60×10^{-1}
Ω_4	1/200	73,856	2.07×10^{-3}	1.01×10^{-1}	60,520	1.49×10^{-3}	7.29×10^{-2}
Rate		–	1.20	1.27	–	1.29	1.13

TABLE 4 Error norms for the shrinking cylinder problem on locally refined meshes.

		Tetrahedral			Hexahedral		
$\Omega(t)$	Δt	#Cells	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$	#Cells	$\Vert\mathbf{u}_h-\mathbf{u}\Vert_{L_2}$	$\Vert p_h-p\Vert_{L_2}$
Ω_1^a	1/25	1104	4.99×10^{-2}	4.82×10^{-1}	384	7.90×10^{-2}	1.47×10^{-0}
Ω_2^a	1/50	3796	1.32×10^{-2}	4.49×10^{-1}	1802	1.45×10^{-2}	3.09×10^{-1}
Ω_3^a	1/100	17,459	4.66×10^{-3}	2.20×10^{-1}	10,449	3.57×10^{-3}	1.48×10^{-1}
Ω_4^a	1/200	101,135	2.05×10^{-3}	9.61×10^{-2}	71,111	1.47×10^{-3}	7.05×10^{-2}
Rate		–	1.18	1.19	–	1.28	1.07

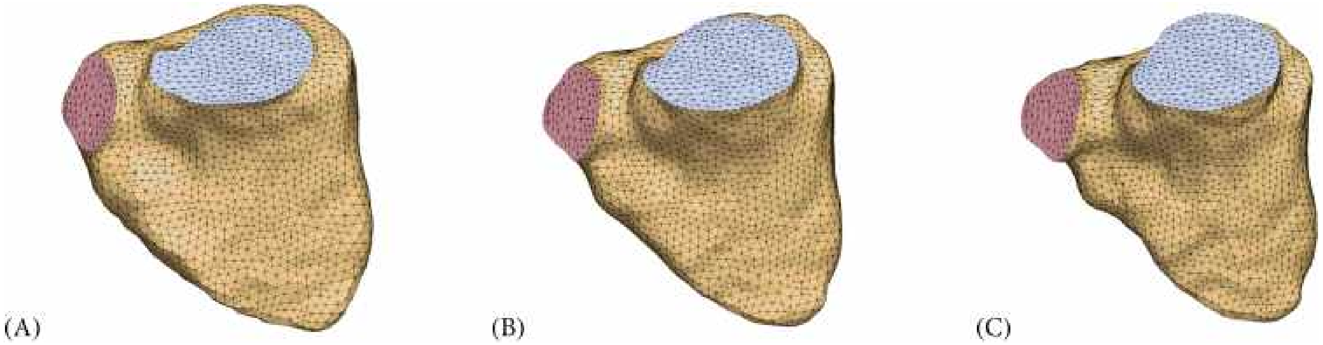

FIGURE 4 The right ventricle surface mesh: (A) beginning of systole (frame 0), (B) middle of systole (frame 23), and (C) end of systole (frame 44).

with a level-set method from the ITK-SNAP package,[109] an initial tetrahedral mesh was constructed by Delaunay triangulation from the CGAL mesh library,[110] and further improved using the aniMBA library from the Ani3D package.[111]

The interval between the mesh frames is assumed to be 10^{-2} (s) that corresponds to the cardiac cycle 9×10^{-1} (s).

For the mesh adaptation, the refinement criterion is based on the absolute vorticity, which is computed in each cell based on the gradient $\mathbf{G}_i$:

$$\operatorname{curl}(\mathbf{u}_i)=\operatorname{curl}(\mathbf{u})|_{\omega_i}=\begin{bmatrix}0&0&0&0&0&0&-1&0&0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0\\0&-1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\mathbf{G}_i. \quad (55)$$

The cells ω_i are marked to be refined if $|\operatorname{curl}(\mathbf{u}_i)|$ exceeds a threshold 10 (s^{-1}). A cell is coarsened if neither of its faces indicates refinement and the coarsening level is not reached.

The initial conditions are zero velocity and pressure. Since these conditions are non-physiological, we perform the simulation for five cardiac cycles that is till $T=4.5$ (s). The maximum time step is $\Delta t=10^{-2}$ (s) and the initial time step is $\Delta t=10^{-5}$ (s).

In this problem, we impose two types of boundary conditions: the no-slip condition Γ_{NS} with $\alpha_\perp = \alpha_\| = 1$, $\beta_\perp = \beta_\| = 0$, $\mathbf{r} = \mathbf{0}$ and the directional-do-nothing condition[46] Γ_{DDN} with $\alpha_\perp = \alpha_\| = \frac{1}{2}(|\mathbf{n}^T\mathbf{u}_1 + n_t| - (\mathbf{n}^T\mathbf{u}_1 + n_t))$, $\beta_\perp = \beta_\| = 1$, $\mathbf{r} = \mathbf{0}$. Note that for the directional-do-nothing condition, the stabilization parameter a_b from (33) is not required.

At the boundary, we distinguish systole surface Γ_s and diastole surface Γ_d. In systolic phase, $t \in [0, 0.45] + 0.9 \times i$ (s), $i \in \mathbb{N}_0$, Γ_s is closed with condition Γ_{NS} and Γ_d is open with condition Γ_{DDN}. In diastolic phase, $t \in [0.45, 0.9) + 0.9 \times i$ (s), $i \in \mathbb{N}_0$, Γ_s is open with condition Γ_{DDN} and Γ_d is closed with condition Γ_{NS}. At $t = 0.45$ (s) both systole and diastole surfaces are open. At the rest of the boundary $\partial\Omega \backslash \Gamma_s \cup \Gamma_d$ the Dirichlet conditions Γ_{NS} are prescribed.

The nonlinear convergence criteria are $\tau_{\mathrm{abs}} = 10^{-4}$, $\tau_{\mathrm{rel}} = 10^{-2}$.

The simulation was performed on the Lomonosov supercomputer.[112,113] We used 25 processors with 14 cores, totaling 350 parallel processes. The mesh was distributed and balanced with the K-means clustering algorithm.[80] The parallel efficiency of the code will be reported elsewhere.

Figures 5 and 6 illustrate the vorticity magnitude $|\operatorname{curl}(\mathbf{u}_i)| \in [0, 25]$ (s^{-1}). Visually, dynamic adaptation to the vorticity magnitude allows to capture more peculiarities of the flow. The corresponding adapted mesh is shown in Figure 7, and mesh partitioning between processors is demonstrated in Figure 8.

The velocity and pressure at various time instants on the dynamically adapted mesh are illustrated in Figures 9 and 10, respectively. Figure 10B corresponds to the moment when both valves are open.

The comparison of maximum velocity and pressure on both variants of meshes (quasi-uniform and locally refined) is shown in Figure 11. Although the velocity is smooth on both grid types, the pressure experiences jumps due to sudden opening and closing of valves.

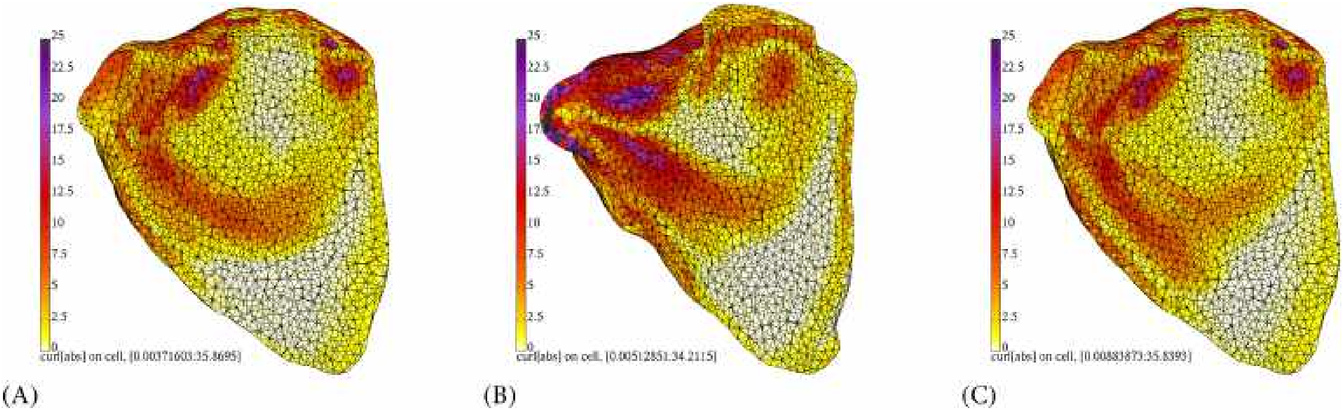
25
22.5
20
17.5
15
12.5
10
7.5
5
2.5
0
curl[abs] on cell, [0.00371603:35.8695]
(A)
curl[abs] on cell, [0.00512851:34.2115]
(B)
curl[abs] on cell, [0.00883873:35.8393]
(C)

FIGURE 5 Cutaway of the quasi-uniform moving mesh, colored by $|\operatorname{curl}(\mathbf{u})| \in [0, 25]$ (s^{-1}) at (A) $t = 3.6$, (B) $t = 4.05$, and (C) $t = 4.5$.

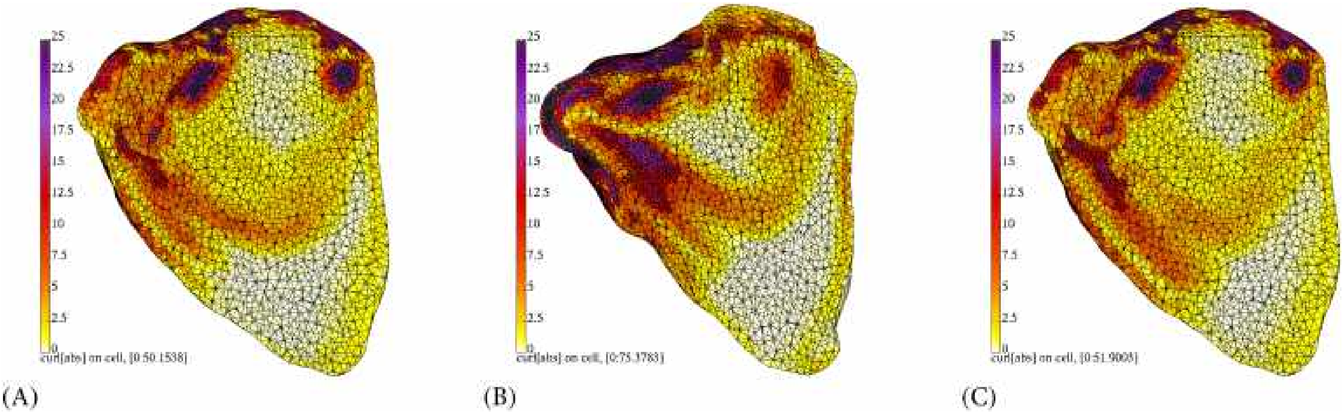
25
22.5
20
17.5
15
12.5
10
7.5
5
2.5
0
curl[abs] on cell, [0:50.1538]
(A)
curl[abs] on cell, [0:75.3783]
(B)
curl[abs] on cell, [0:51.9005]
(C)

FIGURE 6 Cutaway of adaptive moving mesh, colored by $|\operatorname{curl}(\mathbf{u})| \in [0, 25]$ (s^{-1}) at (A) $t = 3.6$, (B) $t = 4.05$, and (C) $t = 4.5$.

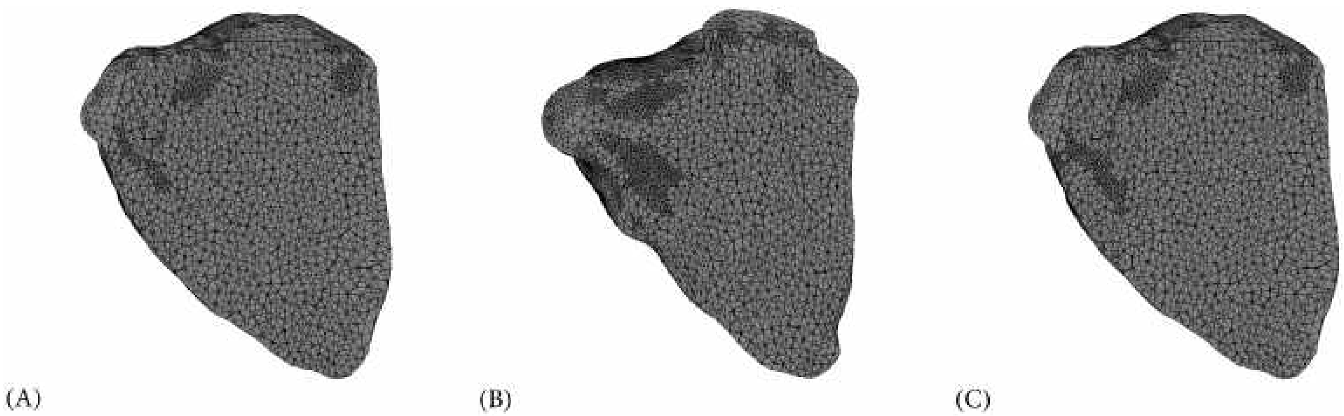

FIGURE 7 Cutaway of adaptive moving mesh at (A) $t = 3.6$ (s), (B) $t = 4.05$ (s), and (C) $t = 4.5$ (s).

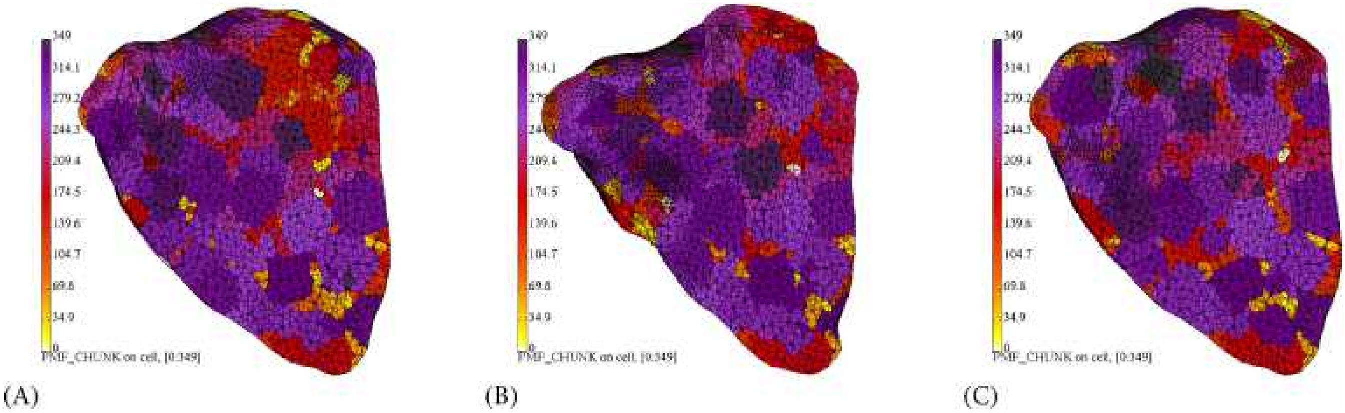

FIGURE 8 Cutaway of adaptive moving mesh, colored in processor number at (A) $t = 3.6$ (s), (B) $t = 4.05$ (s), and (C) $t = 4.5$ (s).

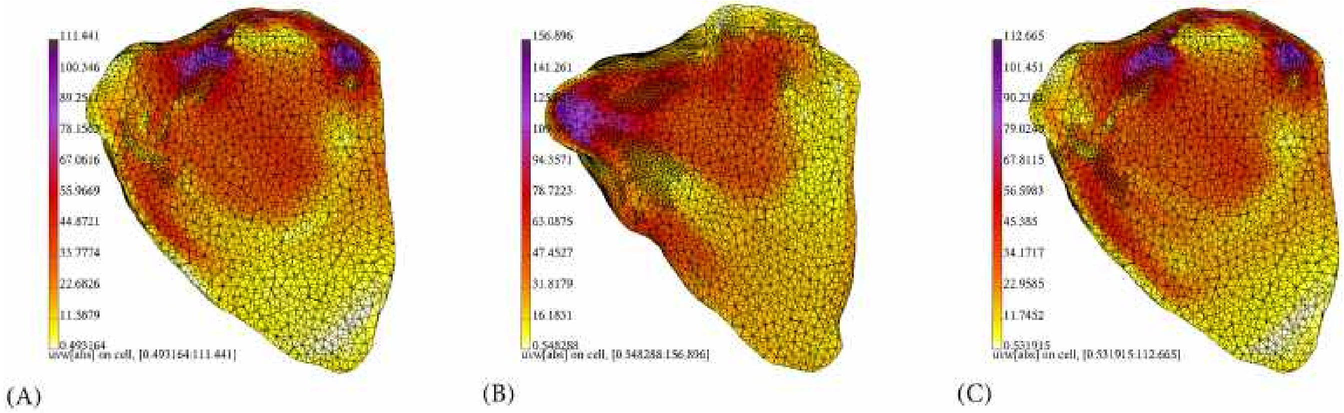

FIGURE 9 Cutaway of adaptive moving mesh, colored in $|\mathbf{u}|$ at (A) $t = 3.6$ (s), (B) $t = 4.05$ (s), and (C) $t = 4.5$ (s).

The evolution of the number of cells for the adaptive mesh is illustrated in Figure 12A. It smoothly follows the physics of the problem and generates almost twice the number of cells of the initial mesh. A few less nonlinear iterations are required on the dynamically adapted mesh, see Figure 12B. The number of nonlinear iterations is quite high in this problem due to pressure spikes occurring during changes of the valve state. The sudden change between systolic and diastolic phases results in convergence issues and cutting of the time step. Nevertheless the method is applicable for patient specific simulations and captures complex peculiarities of the flow using dynamic adaptive moving meshes.

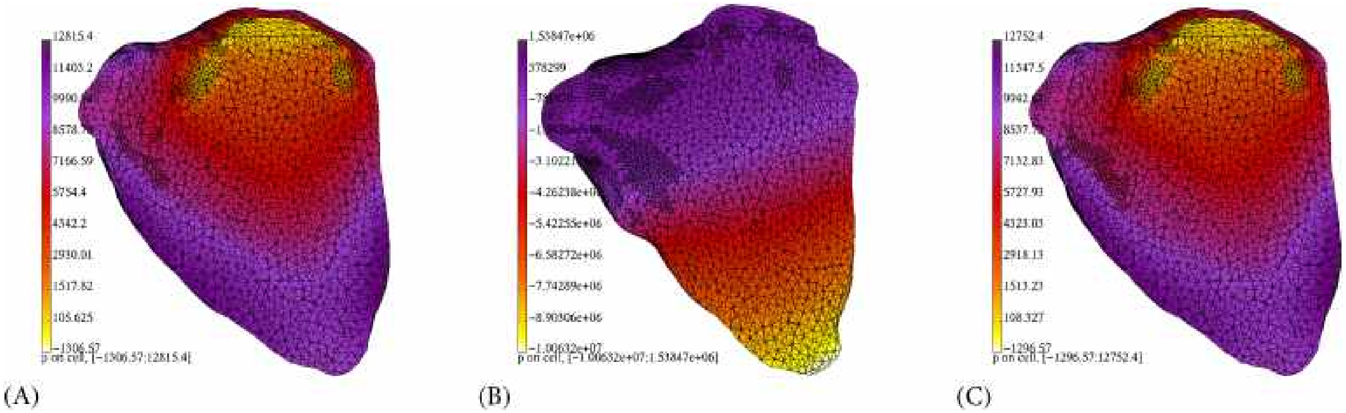

FIGURE 10 Cutaway of adaptive moving mesh, colored in p at (A) $t = 3.6$ (s), (B) $t = 4.05$ (s), and (C) $t = 4.5$ (s).

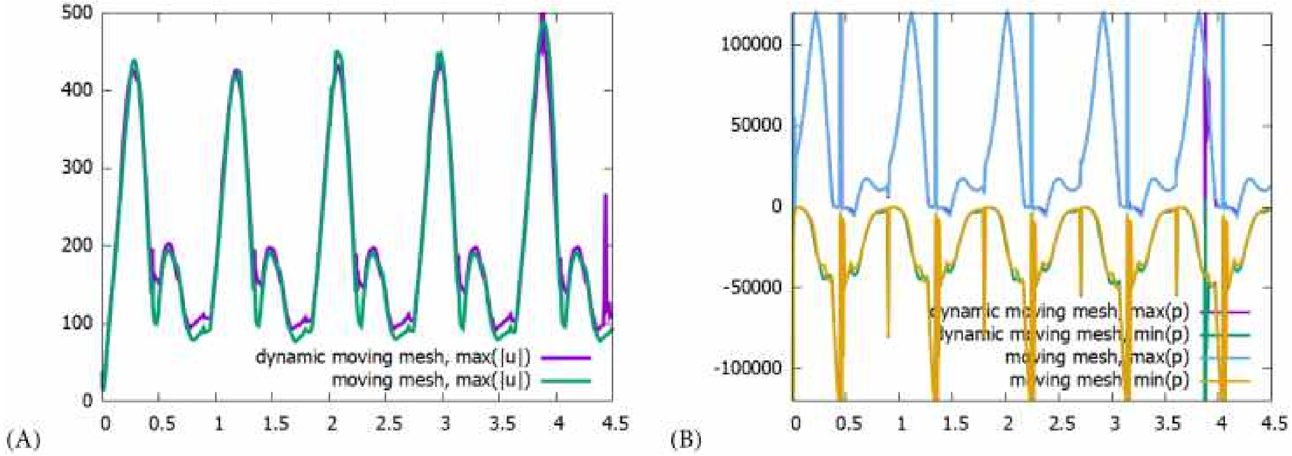

FIGURE 11 Comparison of the solution on dynamically adapted and non-adapted moving mesh (A) maximum velocity and (B) maximum and minimum pressure.

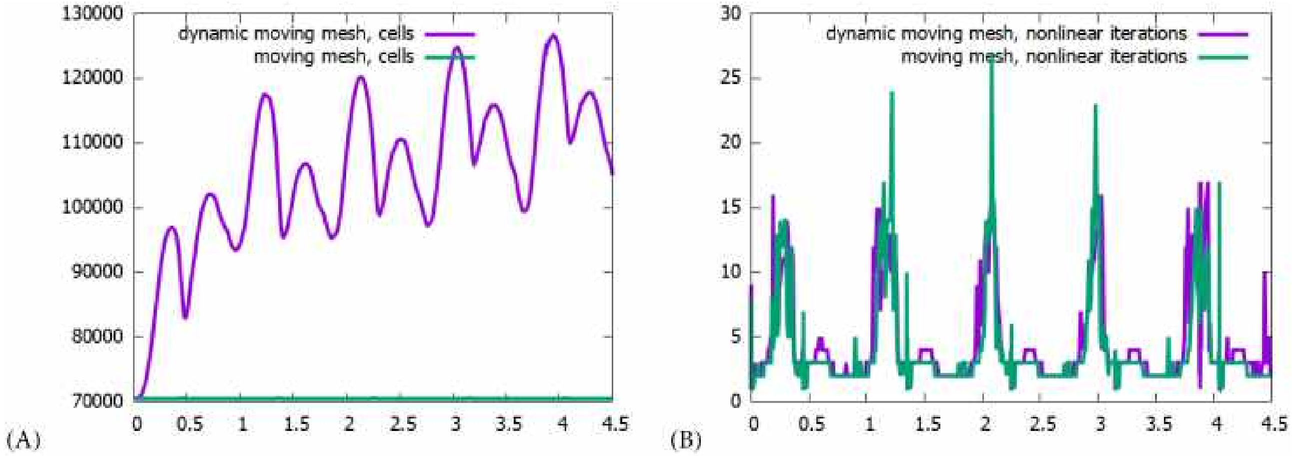

FIGURE 12 Comparison of the solution on dynamically adapted and non-adapted moving mesh (A) number of nonlinear iterations, (B) number of nonlinear iterations per step.

8 | ADVECTION–DIFFUSION–REACTION SYSTEM

In order to model blood coagulation, we augment the system (1) with the additional advection–diffusion–reaction system[1]:

$$
\begin{aligned}
\frac{\partial P}{\partial t}+\operatorname{div}(\mathbf{u}P-D\nabla P)&=-\left(k_1d+k_2B+k_3T+k_4T^2+k_5T^3\right)P,\\
\frac{\partial T}{\partial t}+\operatorname{div}(\mathbf{u}T-D\nabla T)&=\left(k_1d+k_2B+k_3T+k_4T^2+k_5T^3\right)P-k_6AT,\\
\frac{\partial B}{\partial t}+\operatorname{div}(\mathbf{u}B-D\nabla B)&=(k_7d+k_8T)\left(B^0-B\right)-k_9AB,\\
\frac{\partial A}{\partial t}+\operatorname{div}(\mathbf{u}A-D\nabla A)&=-k_6AT-k_9AB,\\
\frac{\partial G}{\partial t}+\operatorname{div}(\mathbf{u}G-D\nabla G)&=-k_{10}TG(k_{11}+G)^{-1},\\
\frac{\partial F}{\partial t}+\operatorname{div}(\mathbf{u}F-D\nabla F)&=k_{10}TG(k_{11}+G)^{-1}-k_{12}F,\\
\frac{\partial M}{\partial t}&=k_{12}F,\\
\frac{\partial f}{\partial t}+\operatorname{div}\left(k(f,d)\left(\mathbf{u}f-D_p\nabla f\right)\right)&=-(k_{13}T+k_{14}d)f,\\
\frac{\partial d}{\partial t}+\operatorname{div}\left(k(f,d)\left(\mathbf{u}d-D_p\nabla d\right)\right)&=(k_{13}T+k_{14}d)f.
\end{aligned}
\tag{56}
$$

Here we denote concentrations of prothrombin P (factor FII), thrombin T (factor FIIa), antithrombin A (factor ATIII), fibrinogen G (factor FI), fibrin F (factor FIa), fibrin polymer M, resting platelets f, activated platelets d, B corresponds to the sum of tissue factors IX and X with $B^0=200$ (nM), The diffusion coefficients are $D=5\times10^{-5}(\mathrm{mm}^2\times\mathrm{s}^{-1})$ for the blood factors and $D_p=2.5\times10^{-5}$ $(\mathrm{mm}^2\times s^{-1})$ for the platelets. The coefficient $k(f,d)=\tan h\left(\pi\phi^{-1}(\phi-f-d)\right)$ limits packing of the platelets to $\phi=400$ $(10^3\times\mathrm{mm}^3)$.[114]

The clotting is accounted in (1) through permeability of the clot that results in the following right-hand side of (1)[115]:

$$
\mathbf{f}=-\nu K^{-1}\mathbf{u},\quad K^{-1}=16r^{-2}S^{3/2}\left(1+56S^3\right)(\phi+d)(\phi-d)^{-1},
\tag{57}
$$

where K is the media permeability, and $S=7000^{-1}\min\,(4900,M)$ is the saturation of the media by fibrin polymer, $r=6\times10^{-4}$ (mm) is the fiber radius. The other coefficients of the model are listed in Table 5.

The initial conditions correspond to the normal pooled plasma: $P_0=1400$, $B_0=10$, $A_0=3400$, $G_0=7000$, $T_0=F_0=M_0=0$ (nM), $f_0=10$, $d_0=0$ $(10^3\ \mathrm{mm}^{-3})$. The boundary conditions for the blood factors $c\in\{P,T,B,A,G,F,M,f,d\}$ are of Dirichlet type $c=c_0$ at the inflow, where c_0 values are taken from the initial conditions. At the rest of the boundary, conditions of the Neumann type $\mathbf{n}\cdot\nabla c=0$ are used for all of the factors. We also distinguish a part of the boundary surface with damaged endothelium, where the tissue factors enter the flow[1]:

$$
D\mathbf{n}\cdot\nabla B=\alpha\left(B^0-B\right)\left(1+\beta\left(B^0-B\right)\right)^{-1},
\tag{58}
$$

$\alpha=7.7\times10^4$ $(\mathrm{nM}^{-1}\ \mathrm{s}^{-1})$ and $\beta=2.25\times10^{-1}$ (nM^{-1}).

8.1 | Approximation of advection–diffusion

We use the finite volume method on the moving mesh for the approximation of the advection–diffusion of the blood factors c, except for the fibrin polymer and platelets. The divergence theorem for the four-integral yields

TABLE 5 Coefficients for the blood coagulation advection–diffusion–reaction system. All the parameters appear in Reference [1].

Coefficient	Value	Units	Coefficient	Value	Units
k_1[1]	1.5×10^{-4}	10^{-3} mm^3 s^{-1}	k_2[116]	7.5×10^{-6}	nM^{-6} s^{-1}
k_3[1]	1.5×10^{-5}	nM^{-1} s^{-1}	k_4[1]	8×10^{-6}	nM^{-2} s^{-1}
k_5[1]	10^{-10}	nM^{-3} s^{-1}	k_6[117]	4.817×10^{-6}	nM^{-1} s^{-1}
k_7[1]	10^{-9}	10^{-3} mm^3 s^{-1}	k_8[118]	5.2173×10^{-5}	nM^{-1} s^{-1}
k_9[117]	2.223×10^{-9}	nM^{-1} s^{-1}	k_{10}[119]	5×10^{-3}	s^{-1}
k_{11}[119]	3160	nM	k_{12}[1]	10^{-1}	s^{-1}
k_{13}[120]	2×10^{-3}	nM^{-1} s^{-1}	k_{14}[114]	4×10^{-9}	10^{-3} mm^3 s^{-1}

$$\int_{\omega(t)} \frac{\partial c}{\partial t} + div(\mathbf{u}c - D\nabla c) \mathrm{d}V(t) = \int_{\omega(t)} \left([\mathbf{u}c - D\nabla c]^T \;\; c \right) \begin{pmatrix} \nabla \\ \partial_t \end{pmatrix} \mathrm{d}V(t) = \oint_{\partial\omega(t)} \left([\mathbf{u}c - D\nabla c]^T \;\; c \right) \mathrm{d}\mathbf{S}(t), \tag{59}$$

the second-order approximation of the integral gives

$$\oint_{\partial\omega(t)} \left([\mathbf{u}c - D\nabla c]^T \;\; c \right) \mathrm{d}\mathbf{S}(t) \approx \sum_{\sigma(t) \in \mathcal{F}(\omega(t))} |\sigma(t)| \left([\mathbf{u}c - D\nabla c]^T \;\; c \right) \begin{pmatrix} \mathbf{n} \\ n_t \end{pmatrix} \Bigg|_{\mathbf{x}_{\sigma(t)}} = \sum_{\sigma(t) \in \mathcal{F}(\omega(t))} |\sigma(t)| \, q_c|_{\mathbf{x}_{\sigma(t)}}, \tag{60}$$

where $q_c = (\mathbf{n}^T\mathbf{u} + n_t)c - D\mathbf{n}^T \nabla c$ is the flux. The velocity projection at the interface $\mathbf{n}^T\mathbf{u}$ is the last component of the coupled flux $\mathbf{F}$ from (24) at the internal faces and (18)–(34) at the boundary faces. Let c_i correspond to concentration c of a blood factor at $\omega_i(t)$ and $\mathbf{g}_i^c = \nabla c|_{\omega_i(t)}$ correspond to the spatial gradient of c at cell $\omega_i(t)$. We assume that c is linear in each cell $\omega_i(t)$ and continuous on the mesh. This allows us to introduce the gradient splitting in cell $\omega_1(t)$ as in (13):

$$\mathbf{g}_1^c = r_1^{-1}\mathbf{n}(c_\sigma - c_1) + \left(\mathbb{I} - r_1^{-1}\mathbf{n}\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \right) \mathbf{g}_1^c. \tag{61}$$

Following the derivation of the flux (18), we obtain the following approximation of q_c at $\mathbf{x}_{\sigma(t)}$ from the side of cell $\omega_1(t)$:

$$q_c|_{\mathbf{x}_{\sigma(t)}} \approx \left(Dr_1^{-1} + s_1\right)c_1 - \left(Dr_1^{-1} + s_1 - \mathbf{n}^T\mathbf{u} - n_t\right)c_\sigma + \left(\left(Dr_1^{-1} + s_1\right)\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T - D\mathbf{n}^T \right) \mathbf{g}_1^c, \tag{62}$$

where $s_1 = \max\left(\mathbf{n}^T\mathbf{u} + n_t - Dr_1^{-1}, 0\right)$ is the stabilization parameter. Considering approximation from the side of the adjacent cell $\omega_2(t)$ we get:

$$q_c|_{\mathbf{x}_{\sigma(t)}} \approx \left(Dr_2^{-1} + s_2 + \mathbf{n}^T\mathbf{u} + n_t\right)c_\sigma - \left(Dr_2^{-1} + s_2\right)c_2 - \left(\left(Dr_2^{-1} + s_2\right)\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T + D\mathbf{n}^T \right) \mathbf{g}_2^c, \tag{63}$$

with $s_2 = \max\left(-\mathbf{n}^T\mathbf{u} - n_t - Dr_2^{-1}, 0\right)$.

Equating the two approximations (62) and (63) we obtain c_σ and the flux approximation:

$$\begin{aligned} q_c|_{\mathbf{x}_{\sigma(t)}} &\approx \left(Dr_2^{-1} + s_2 + \mathbf{n}^T\mathbf{u} + n_t\right)\left(D\left(r_1^{-1} + r_2^{-1}\right) + s_1 + s_2\right)^{-1} \left(\left(Dr_1^{-1} + s_1\right)\left(c_1 + \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \mathbf{g}_1^c\right) - D\mathbf{n}^T\mathbf{g}_1^c \right) \\ &- \left(Dr_1^{-1} + s_1 - \mathbf{n}^T\mathbf{u} - n_t\right)\left(D\left(r_1^{-1} + r_2^{-1}\right) + s_1 + s_2\right)^{-1} \left(\left(Dr_2^{-1} + s_2\right)\left(c_2 + \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T \mathbf{g}_2^c\right) + D\mathbf{n}^T\mathbf{g}_2^c \right). \end{aligned} \tag{64}$$

At the boundary of the Dirichlet type, the flux is approximated by using $c_\sigma = c_0$ in (62). At the boundary of the Neumann type, we split the gradient using (61) to obtain c_σ that can be used in (62):

$$c_\sigma = c_1 + \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)} - r_1\mathbf{n}\right)^T \mathbf{g}_1^c. \tag{65}$$

Finally, for the nonlinear boundary condition (58) we apply the Taylor series expansion for the right hand side:

$$\alpha\left(B^0 - B\right)\left(1+\beta\left(B^0 - B\right)\right)^{-1}\Big|_{\mathbf{x}_{\sigma(t)}} \approx \alpha\left(B^0 - B_1\right)\left(1+\beta\left(B^0 - B_1\right)\right)^{-1} - \alpha\left(1+\beta\left(B^0 - B_1\right)\right)^{-2}\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)\cdot\nabla B, \tag{66}$$

where B_1 and B_σ are the concentrations of tissue factors at cell ω_1 and face σ, respectively. Splitting the gradients using (61) in (58) and (66) we obtain B_σ:

$$B_\sigma = B_1 + \left(\alpha\left(1+\beta\left(B^0 - B_1\right)\right)^{-2} + Dr_1^{-1}\right)^{-1}\left(\alpha\left(B^0 - B_1\right)\left(1+\beta\left(B^0 - B_1\right)\right)^{-1} + Dr_1^{-1}\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)} - r_1\mathbf{n}\right)^T\mathbf{g}_1^B\right), \tag{67}$$

which is used in (62) to obtain the boundary flux for the tissue factors IX, X at the surface of damaged endothelium.

8.2 | Approximation of platelets' advection–diffusion

The platelets' advection–diffusion contains a nonlinear coefficient $k(f,d)$. This term couples the two equations for the platelets, thus we integrate them together:

$$\int_{\omega(t)} \frac{\partial}{\partial t}\begin{bmatrix} f \\ d \end{bmatrix} + \operatorname{div}\left(k(f,d)\begin{bmatrix}\mathbf{u}f - D_p\nabla f \\ \mathbf{u}d - D_p\nabla d\end{bmatrix}\right)\mathrm{d}V(t) = \int_{\omega(t)}\left(k(f,d)\begin{bmatrix}\mathbf{u}^T f - D_p\nabla f^T \\ \mathbf{u}^T d - D_p\nabla d^T\end{bmatrix}\ \begin{bmatrix} f \\ d\end{bmatrix}\right)\begin{pmatrix}\nabla \\ \partial_t\end{pmatrix}\mathrm{d}V(t), \tag{68}$$

and apply the divergence theorem and the second-order approximation to get:

$$\begin{aligned}
&\oint_{\partial\omega(t)}\left(k(f,d)\begin{bmatrix}\mathbf{u}^T f - D_p\nabla f^T \\ \mathbf{u}^T d - D_p\nabla d^T\end{bmatrix}\ \begin{bmatrix} f \\ d\end{bmatrix}\right)\mathrm{d}\mathbf{S}(t) \approx \sum_{\sigma(t)\in\mathcal{F}(\omega(t))} |\sigma(t)|\left(k(f,d)\begin{bmatrix}\mathbf{u}^T f - D_p\nabla f^T \\ \mathbf{u}^T d - D_p\nabla d^T\end{bmatrix}\ \begin{bmatrix} f \\ d\end{bmatrix}\right)\begin{pmatrix}\mathbf{n} \\ n_t\end{pmatrix} \\
&= \sum_{\sigma(t)\in\mathcal{F}(\omega(t))} |\sigma(t)|\left(\left(k(f,d)\mathbf{n}^T\mathbf{u} + n_t\right)\begin{bmatrix} f \\ d\end{bmatrix} - k(f,d)D_p\begin{bmatrix}\mathbf{n}^T\nabla f \\ \mathbf{n}^T\nabla d\end{bmatrix}\right) = \sum_{\sigma(t)\in\mathcal{F}(\omega(t))} |\sigma(t)|\,\mathbf{q}_p,
\end{aligned} \tag{69}$$

where $\mathbf{q}_p$ is the coupled advection–diffusion flux for the platelets. We first note that

$$k'(f,d) = \partial_f k(f,d) = \partial_d k(f,d) = -\pi p^{-1}\operatorname{sech}\left(\pi p^{-1}(p - f - d)\right)^2, \quad \partial_d\nabla d = \partial_f\nabla f = \nabla 1 = \mathbf{0}, \tag{70}$$

approximate $\mathbf{q}_p$ from the side of cell $\omega_1(t)$ at face $\sigma(t)$ using the Taylor series expansion

$$\begin{aligned}
\mathbf{q}_p\Big|_{\mathbf{x}_{\sigma(t)}} &\approx \left(\left(k(f,d)\mathbf{n}^T\mathbf{u} + n_t\right)\begin{bmatrix} f \\ d\end{bmatrix} - k(f,d)D_p\begin{bmatrix}\mathbf{n}^T\nabla f \\ \mathbf{n}^T\nabla d\end{bmatrix}\right)\Bigg|_{\mathbf{x}_{\omega_1(t)}} \\
&+ \left(\left(k(f,d)\mathbf{n}^T\mathbf{u} + n_t\right)\mathbb{I} + k'(f,d)\begin{bmatrix}\mathbf{n}^T\mathbf{u}f - D_p\mathbf{n}^T\nabla f \\ \mathbf{n}^T\mathbf{u}d - D_p\mathbf{n}^T\nabla d\end{bmatrix}\mathbf{1}^T\right)\Bigg|_{\mathbf{x}_{\omega_1(t)}} \otimes\left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T\left(\begin{bmatrix} f \\ d\end{bmatrix}\otimes\nabla\right)\Bigg|_{\mathbf{x}_{\omega_1(t)}},
\end{aligned} \tag{71}$$

and proceed with the gradient decomposition (61).

$$\begin{aligned}\mathbf{q}_p\Big|_{\mathbf{x}_{\sigma(t)}} &\approx \left(\Lambda_1^p + \left(k(f_1,d_1)\mathbf{n}^T\mathbf{u} + n_t\right)\mathbb{I}\right)\begin{bmatrix} f_1 \\ d_1 \end{bmatrix} - \Lambda_1^p \begin{bmatrix} f_\sigma \\ d_\sigma \end{bmatrix} \\ &-k(f_1,d_1)D_p\begin{bmatrix} \mathbf{n}^T\mathbf{g}_1^f \\ \mathbf{n}^T\mathbf{g}_1^d \end{bmatrix} + \left(k(f_1,d_1)D_p r_1^{-1} + s_1^p\right)\begin{bmatrix} \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^f \\ \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^d \end{bmatrix},\end{aligned} \tag{72}$$

where s_1^p is a stabilization parameter, and Λ_1^p is a 2×2 matrix coefficient:

$$\Lambda_1^p = \left(k(f_1,d_1)D_p r_1^{-1} + s_1^p - k(f_1,d_1)\mathbf{n}^T\mathbf{u} - n_t\right)\mathbb{I} - k'(f_1,d_1)\begin{bmatrix} \mathbf{n}^T\mathbf{u}f_1 - D_p\mathbf{n}^T\mathbf{g}_1^f \\ \mathbf{n}^T\mathbf{u}d_1 - D_p\mathbf{n}^T\mathbf{g}_1^d \end{bmatrix}\mathbf{1}^T. \tag{73}$$

The eigenvalues of Λ_1^p are:

$$\lambda_1 = k(f_1,d_1)D_p r_1^{-1} + s_1^p - k(f_1,d_1)\mathbf{n}^T\mathbf{u} - n_t, \quad \lambda_2 = \lambda_1 - k'(f_1,d_1)\left(\mathbf{n}^T\mathbf{u}(f_1+d_1) - D_p\mathbf{n}^T\left(\mathbf{g}_1^f + \mathbf{g}_1^d\right)\right), \tag{74}$$

whereas the eigenvalues of $\Lambda_1^p + (k(f_1,d_1)\mathbf{n}^T\mathbf{u} + n_t)\mathbb{I}$ are:

$$\lambda_1 = k(f_1,d_1)D_p r_1^{-1} + s_1^p, \quad \lambda_2 = \lambda_1 - k'(f_1,d_1)\left(\mathbf{n}^T\mathbf{u}(f_1+d_1) - D_p\mathbf{n}^T\left(\mathbf{g}_1^f + \mathbf{g}_1^d\right)\right), \tag{75}$$

which results in the following choice for the stabilization parameter s_1^p:

$$s_1^p = \max\left(\max\left(k'(f_1,d_1)\left(\mathbf{n}^T\mathbf{u}(f_1+d_1) - D_p\mathbf{n}^T\left(\mathbf{g}_1^f + \mathbf{g}_1^d\right)\right),0\right) + \max\left(k(f_1,d_1)\mathbf{n}^T\mathbf{u} + n_t,0\right) - k(f_1,d_1)D_p r_1^{-1},0\right). \tag{76}$$

The flux approximation from an adjacent cell $\omega_2(t)$ that shares the same face $\sigma(t) = \omega_1(t)\cap\omega_2(t)$ reads as:

$$\begin{aligned}\mathbf{q}_p\Big|_{\mathbf{x}_{\sigma(t)}} &\approx \Lambda_2^p\begin{bmatrix} f_\sigma \\ d_\sigma \end{bmatrix} - \left(\Lambda_2^p - \left(k(f_2,d_2)\mathbf{n}^T\mathbf{u} + n_t\right)\mathbb{I}\right)\begin{bmatrix} f_2 \\ d_2 \end{bmatrix} \\ &-k(f_2,d_2)D_p\begin{bmatrix} \mathbf{n}^T\mathbf{g}_2^f \\ \mathbf{n}^T\mathbf{g}_2^d \end{bmatrix} - \left(k(f_2,d_2)D_p r_2^{-1} + s_2^p\right)\begin{bmatrix} \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T\mathbf{g}_2^f \\ \left(\mathbf{x}_{\sigma(t)} - \mathbf{x}_{\omega_2(t)}\right)^T\mathbf{g}_2^d \end{bmatrix},\end{aligned} \tag{77}$$

with Λ_2^p given by

$$\Lambda_2^p = \left(k(f_2,d_2)D_p r_2^{-1} + s_2^p + k(f_2,d_2)\mathbf{n}^T\mathbf{u} + n_t\right)\mathbb{I} + k'(f_2,d_2)\begin{bmatrix} \mathbf{n}^T\mathbf{u}f_2 - D_p\mathbf{n}^T\mathbf{g}_2^f \\ \mathbf{n}^T\mathbf{u}d_2 - D_p\mathbf{n}^T\mathbf{g}_2^d \end{bmatrix}\mathbf{1}^T, \tag{78}$$

and the choice for s_2^p is

$$s_2^p = \max\left(\max\left(-k'(f_2,d_2)\left(\mathbf{n}^T\mathbf{u}(f_2+d_2) - D_p\mathbf{n}^T\left(\mathbf{g}_2^f + \mathbf{g}_2^d\right)\right),0\right) + \max\left(-k(f_2,d_2)\mathbf{n}^T\mathbf{u} - n_t,0\right) - k(f_2,d_2)D_p r_2^{-1},0\right). \tag{79}$$

Equating (73) and (77) we eliminate interface unknowns and obtain the flux expression:

$$\begin{aligned}\mathbf{q}_p\Big|_{\mathbf{x}_{\sigma(t)}} \approx\ & \Lambda_2^p(\Lambda_1^p+\Lambda_2^p)^{-1}\left(\left(\Lambda_1^p+\left(k(f_1,d_1)\mathbf{n}^T\mathbf{u}+n_t\right)\mathbb{I}\right)\begin{bmatrix} f_1 \\ d_1\end{bmatrix}+\left(k(f_1,d_1)\ D_p r_1^{-1}+s_1^p\right)\begin{bmatrix}\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^f \\ \left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^d\end{bmatrix}\right) \\ & -\Lambda_1^p(\Lambda_1^p+\Lambda_2^p)^{-1}\left(\left(\Lambda_2^p-\left(k(f_2,d_2)\mathbf{n}^T\mathbf{u}+n_t\right)\mathbb{I}\right)\begin{bmatrix} f_2 \\ d_2\end{bmatrix}+\left(k(f_2,d_2)\ D_p r_2^{-1}+s_2^p\right)\begin{bmatrix}\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_2(t)}\right)^T\mathbf{g}_2^f \\ \left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_2(t)}\right)^T\mathbf{g}_2^d\end{bmatrix}\right) \\ & -k(f_1,d_1)\ D_p\Lambda_2^p(\Lambda_1^p+\Lambda_2^p)^{-1}\begin{bmatrix}\mathbf{n}^T\mathbf{g}_1^f \\ \mathbf{n}^T\mathbf{g}_1^d\end{bmatrix}-k(f_2,d_2)\ D_p\Lambda_1^p(\Lambda_1^p+\Lambda_2^p)^{-1}\begin{bmatrix}\mathbf{n}^T\mathbf{g}_2^f \\ \mathbf{n}^T\mathbf{g}_2^d\end{bmatrix}.\end{aligned} \tag{80}$$

Finally, at Dirichlet and Neumann boundaries, we substitute interface unknowns in (73) by

$$\begin{bmatrix} f_\sigma \\ d_\sigma\end{bmatrix}=\begin{bmatrix} f_0 \\ d_0\end{bmatrix},\quad \text{and}\quad \begin{bmatrix} f_\sigma \\ d_\sigma\end{bmatrix}=\begin{bmatrix} f_1+\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}-r_1\mathbf{n}\right)^T\mathbf{g}_1^f \\ d_1+\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}-r_1\mathbf{n}\right)^T\mathbf{g}_1^d\end{bmatrix}, \tag{81}$$

respectively.

8.3 | Gradient reconstruction

We reconstruct the concentration gradient $\mathbf{g}_1^c$ at the center of each cell $\omega_i(t)\in\Omega(t_{n+1})$. Consider a cell ω_1, for every other cell sharing any element $\omega_2(t)\in\mathcal{V}(\Omega(t_{n+1}))$, $\omega_1\cap\omega_2\neq\emptyset$, $\omega_2\neq\omega_1$, we consider the following condition for the gradient:

$$\left(\mathbf{x}_{\omega_2(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^c=c_2-c_1. \tag{82}$$

For boundary faces $\sigma(t)\in\mathcal{F}(\partial\Omega(t))$, $\sigma(t)=\partial\Omega(t)\cap\omega_1(t)$, $\omega_1(t)\in\mathcal{V}(\Omega(t_{n+1}))$ with Dirichlet and Neumann conditions, we get

$$\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\mathbf{g}_1^c=c_0-c_1,\quad \mathbf{n}^T\mathbf{g}_1^c=0, \tag{83}$$

respectively. For boundary faces with nonlinear boundary condition (58) the Taylor expansion (66) results in the following condition for the gradient $\mathbf{g}_1^B$ of tissue factors B:

$$\left(\alpha^{-1}\left(1+\beta\left(B^0-B_1\right)\right)^2D\mathbf{n}^T+\left(\mathbf{x}_{\sigma(t)}-\mathbf{x}_{\omega_1(t)}\right)^T\right)\mathbf{g}_1^B=\left(B^0-B_1\right)\left(1+\beta\left(B^0-B_1\right)\right). \tag{84}$$

Gathering conditions (82), (83), and (84) from all faces of a cell $\omega_1(t)$ we obtain a system $A_c\mathbf{g}_1^c=b_c$, solved with the Cholesky method: $\mathbf{g}_1^c=\left(A_c^TA_c\right)^{-1}A_c^Tb_c$. Note that the contribution of (84) into A_c is nonlinear.

Similarly to Section 3.5, the gradient stencil includes cells sharing at least a node and may appear too wide for some configurations that may degrade the performance of the solver.

8.4 | Approximation of reactions

We use a matrix-weighted Euler method for time-integration of a reaction system.[4] To this end, let us consider the initial value problem:

$$\frac{\partial\mathbf{x}}{\partial t}=\mathbf{r}(\mathbf{x}),\quad \mathbf{x}(0)=\mathbf{x}_0,\quad \mathbf{x}\in\mathbb{R}^d, \tag{85}$$

where d is the problem dimension and function $\mathbf{r}(\mathbf{x}) \in C^1 : \mathbb{R}^d \to \mathbb{R}^d$.

In this work, $\mathbf{x}$ corresponds to the blood factors in $\mathbf{x}$ and $\mathbf{r}(\mathbf{x})$ to the right hand sides from (56):

$$\mathbf{x} = \begin{bmatrix} P \\ T \\ B \\ A \\ G \\ F \\ M \\ f \\ d \end{bmatrix}, \quad \mathbf{r}(\mathbf{x}) = \begin{bmatrix} -(k_1 d + k_2 B + k_3 T + k_4 T^2 + k_5 T^3)P \\ (k_1 d + k_2 B + k_3 T + k_4 T^2 + k_5 T^3)P - k_6 AT \\ (k_7 d + k_8 T)(B^0 - B) - k_9 AB \\ -k_6 AT - k_9 AB \\ -k_{10} TG(k_{11} + G)^{-1} \\ k_{10} TG(k_{11} + G)^{-1} - k_{12} F \\ k_{12} F \\ -(k_{13} T + k_{14} d)f \\ (k_{13} T + k_{14} d)f \end{bmatrix}. \tag{86}$$

Integration of (85) over space–time leads to the following approximation of (85) in cell ω_i:

$$\int_{\omega(t)} \left(\frac{\partial \mathbf{x}}{\partial t} - \mathbf{r}(\mathbf{x}) \right) \mathrm{d}V(t) \approx |\omega^{n+1}|\,\mathbf{x}^{n+1} - |\omega^n|\,\mathbf{x}^n - |\omega_i(t_n, t_{n+1})|\,\mathbf{r}(\mathbf{x}^*), \tag{87}$$

where choices $\mathbf{x}^* \equiv \mathbf{x}^n$ and $\mathbf{x}^* \equiv \mathbf{x}^{n+1}$ correspond to the forward and backward Euler methods, respectively. We solve (85) with the matrix-weighted combination of the forward and backward Euler methods, which leads to the following residual for (85):

$$\mathcal{R}\left(\mathbf{x}^{n+1}\right) = |\omega^{n+1}|\,\mathbf{x}^{n+1} - |\omega^n|\,\mathbf{x}^n - |\omega_i(t_n, t_{n+1})|\left(W\mathbf{r}\left(\mathbf{x}^{n+1}\right) + (\mathbb{I} - W)\mathbf{r}(\mathbf{x}^n)\right) = \mathbf{0}, \tag{88}$$

where the matrix weight $W \in \mathbb{R}^{d \times d}$ depends on the generalized weight function $\theta(z) = z^{-1} - (e^z - 1)^{-1} : \mathbb{C} \to \mathbb{C}$ and Jacobian matrix $\mathbf{J}(\mathbf{x}) = \partial \mathbf{r}(\mathbf{x}) / \partial \mathbf{x}^T \in \mathbb{R}^{d \times d}$ by $W = \theta(\xi \mathbf{J})$, with $\xi = |\omega_i(t_n, t_{n+1})| / |\omega_i^{n+1}|$.

In order to compute $\theta(\xi \mathbf{J})$, we approximate $\theta(z)$ using the Taylor series expansion for the positive and negative arguments of the exponent:

$$\theta(z) \approx \theta^+\left(\frac{z + |z|}{2}\right) + \theta^-\left(\frac{z - |z|}{2}\right) - \frac{1}{2}, \tag{89}$$

where $\theta^+(z)$ and $\theta^-(z)$ are given by:

$$\theta^+(z) = \left(120\left(60 + 20z + 5z^2 + z^3\right)^{-1} + z\right)^{-1}, \quad \theta^-(z) = 1 - \left(120\left(60 - 20z + 5z^2 - z^3\right)^{-1} - z\right)^{-1}. \tag{90}$$

and split $\mathbf{J} = \mathbf{J}_+ + \mathbf{J}_-$, $\mathbf{J}_\pm = \left(\mathbf{J} \pm \sqrt{\mathbf{J}^T \mathbf{J}}\right)/2$, with the matrix root computed using the singular value decomposition. Thus, the matrix weight is

$$W = \theta^+(\xi \mathbf{J}_+) + \theta^-(\xi \mathbf{J}_-) - \mathbb{I}/2. \tag{91}$$

9 | SOLUTION OF THE COUPLED PROBLEM

At the beginning of the Newton iterations, we select the initial guess $\mathbf{x}_0^{n+1} = \mathbf{x}^n$ and evaluate $\mathbf{r}^n = \mathbf{r}(\mathbf{x}^n)$. At the kth Newton step of the coupled problem for blood flow and coagulation, we assemble the residual as follows:

1. Compute the residual (88) for the reactions $\mathcal{R}^k$ at every cell $\omega_i(t_{n+1})$
   - Evaluate the function $\mathbf{r}_k^{n+1} = \mathbf{r}\left(\mathbf{x}_k^{n+1}\right)$ and the Jacobian $\mathbf{J}_k = \partial \mathbf{r}(\mathbf{x}) / \partial \mathbf{x}^T|_{\mathbf{x}=\mathbf{x}_k^{n+1}}$.
   - Compute W according to (91) and add (88) to residual $\mathcal{R}_i^k$, corresponding to the blood factors. The residual for velocity and pressure remains zero.
   - Add the temporal boundary fluxes (35) and subtract the right-hand side (57) (evaluated with unknowns at the current iteration) from the residual vector, corresponding to velocity.
2. Compute the gradients $\mathbf{G}_i$ and $\mathbf{g}_i^c$ with derivatives at every cell $\omega_i(t_{n+1})$ following sections 3.5 and 8.3, respectively.
3. Add the divergent part to the residual at every cell $\omega_i(t) \in \mathcal{V}(\Omega(t_{n+1}))$:
   - Add $\sum_{\sigma(t)} |\sigma(t)| \mathbf{F}|_{\mathbf{x}_{\sigma(t)}}$ with the coupled flux $\mathbf{F}$ computed by (24) and (18)–(34) to the part of the residual $\mathcal{R}_i^k$ corresponding to velocity $\mathbf{u}$ and pressure p.
   - Add $\sum_{\sigma(t)} |\sigma(t)| q_c|_{\mathbf{x}_{\sigma(t)}}$ of every blood factor c (except for fibrin-polymer M and platelets f, d) to the corresponding part of the residual $\mathcal{R}_i^k$.
   - Add $\sum_{\sigma(t)} |\sigma(t)| \mathbf{q}_p|_{\mathbf{x}_{\sigma(t)}}$ to the part of the residual $\mathcal{R}_i^k$, corresponding to platelets f and d.

Here again, for the refinement, we reuse gradients $\mathbf{g}_i^c$, computed at the last Newton iteration. During the refinement step, the interpolation from a coarse cell $\omega_1(t)$ to fine cells $\omega_i(t)$ is computed by

$$c_i = c_i + \theta_c \left(\mathbf{x}_{\omega_i(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \mathbf{g}_1^c, \quad \mathbf{g}_i^c = \mathbf{g}_1^c, \tag{92}$$

where θ_c is chosen to limit the interpolation for all new cells $\omega_i(t)$:

$$\min_{\omega_j(t) \in \mathcal{V}_n(\omega_1(t))} (c_j) \le c_1 + \theta_c \left(\mathbf{x}_{\omega_i(t)} - \mathbf{x}_{\omega_1(t)}\right)^T \mathbf{g}_1^c \le \max_{\omega_j(t) \in \mathcal{V}_n(\omega_1(t))} (c_j), \tag{93}$$

$\mathcal{V}_n(\omega_1(t))$ is a set of cells sharing at least a node with the cell $\omega_1(t)$. The interpolation is conservative under condition $\sum_i |\omega_i(t)| \mathbf{x}_i = |\omega_1(t)| \mathbf{x}_1$ and is monotone due to (93). During coarsening of fine cells ω_i to a coarse cell ω_1 we use the simple averaging:

$$c_1 = |\omega_1(t)|^{-1} \sum_i |\omega_i(t)| c_i, \quad \mathbf{g}_1^c = |\omega_1(t)|^{-1} \sum_i |\omega_i(t)| \mathbf{g}_i^c, \tag{94}$$

which is both monotone and conservative.

10 | COAGULATION IN STATIC AND PULSATING CAPILLARIES

We reproduce the numerical simulation[1] of blood coagulation in static microfluidic capillaries[2] and extend virtually the experiment to pulsating capillaries. The original domain Ω is represented by the cylinder of length $L = 8$ $[mm]$ and radius $R = 1$ $[mm]$ aligned with z axis:

$$\Omega = \left\{\mathbf{x} = (x, y, z) : 0 \ [mm] \le z \le L, \ \ x^2 + y^2 \le R^2\right\}. \tag{95}$$

At the outflow $\partial\Omega|_{z=8}$, we prescribe the directional do-nothing condition Γ_{DDN} with $\alpha_\perp = \alpha_\parallel = \frac{1}{2}(|\mathbf{n}^T\mathbf{u}_1 + n_t| - (\mathbf{n}^T\mathbf{u}_1 + n_t))$, $\beta_\perp = \beta_\parallel = 1$, and $\mathbf{r} = \mathbf{0}$. At the inflow $\partial\Omega|_{z=0}$, we prescribe the directional pressure boundary condition Γ_{DP} with $\alpha_\perp = \alpha_\parallel = \frac{1}{2}(|\mathbf{n}^T\mathbf{u}_1 + n_t| - (\mathbf{n}^T\mathbf{u}_1 + n_t))$, $\beta_\perp = \beta_\parallel = 1$, and $\mathbf{r} = -p_{in}\mathbf{n}$. At the rest of the boundary $\partial\Omega \backslash \left(\partial\Omega|_{z=0} \cup \partial\Omega|_{z=8}\right)$, the no-slip conditions Γ_{NS} are prescribed with $\alpha_\perp = \alpha_\parallel = 1$, $\beta_\perp = \beta_\parallel = 0$ and $\mathbf{r} = \mathbf{0}$.

The initial conditions are: zero velocity and pressure. The initial conditions, boundary conditions, and model coefficients for the blood factors are described in Section 8, see also Figure 13. The maximum time step is $\Delta t = 1/4$ $[s]$, and the initial time step is 10^{-3} $[s]$.

The inflow pressure is dictated by the target shear rate γ by the formula $p_{in} = 2\nu\gamma L R^{-1}$. Here we consider pressure providing maximum analytical velocity for the flow in the cylindrical pipe 5 $[mm \cdot s^{-1}]$ and the shear rate $\gamma = 20$ $[s^{-1}]$.

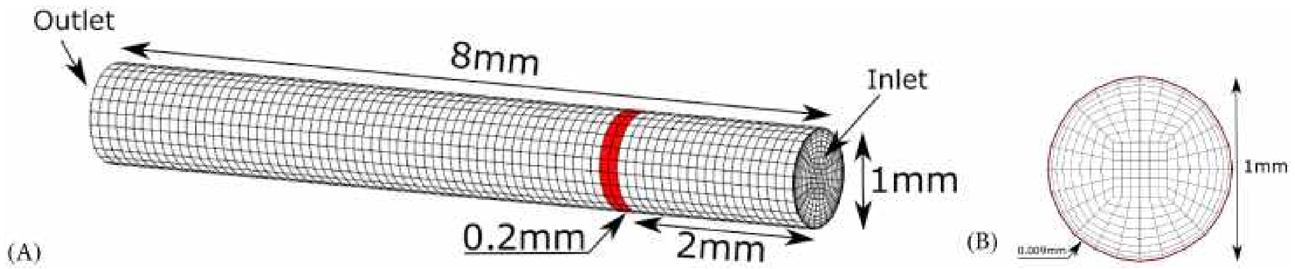

FIGURE 13 Problem setup on the initial mesh for the experiment of blood flow and coagulation in microfluidic capillaries (A) with the aggressive refinement to the cylinder boundary (B). The red stripe in (A) corresponds to the area with damaged endothelium.

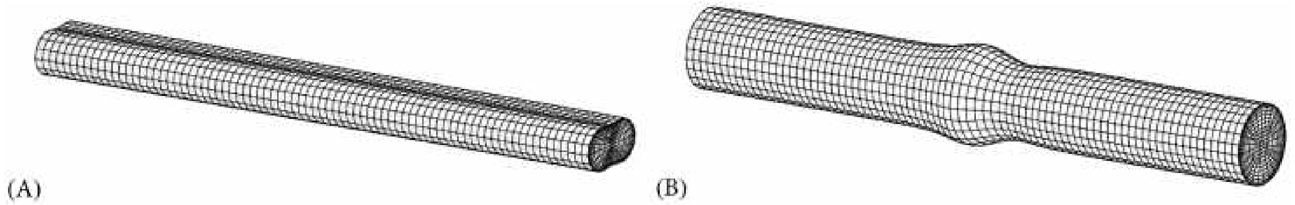

FIGURE 14 Dumbbell-shaped deformation (A) and pulse-wave deformation (B) at $t = 8$ (s). The outflow is to the left.

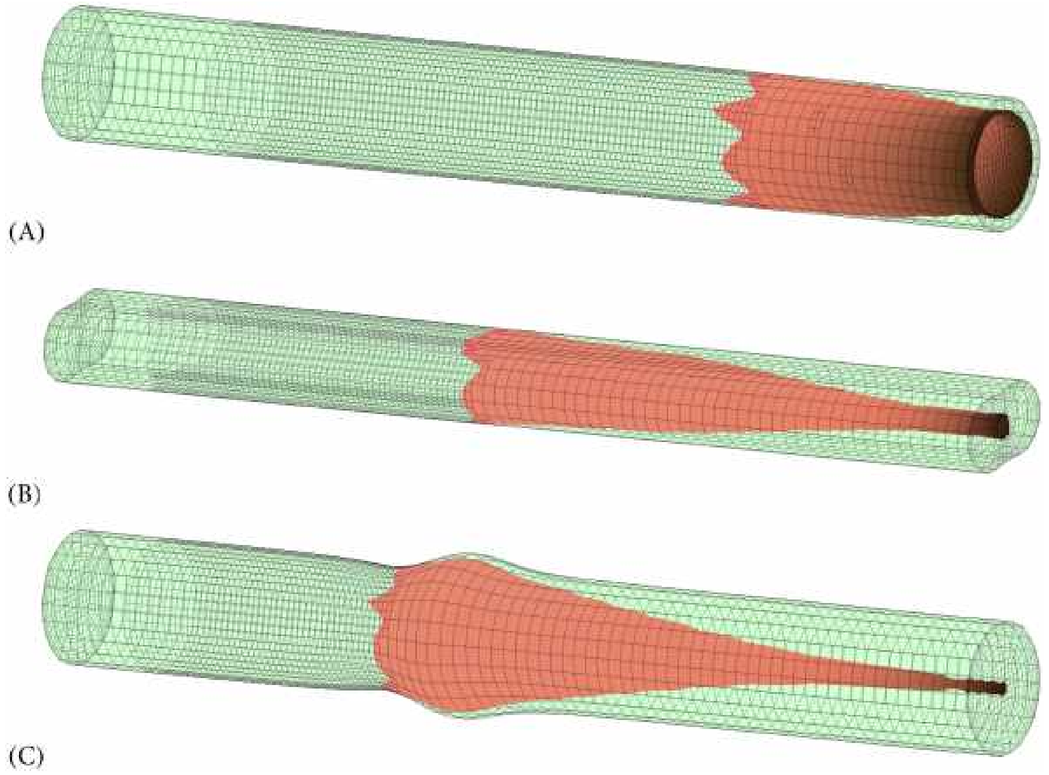

FIGURE 15 Side view of K^{-1}-isosurface 100 (mm^{-2}) at $t = 87$ (s) for (A) static capillary, (B) dumbbell-shaped deformation and (C) pulse-wave deformation. The isosurface is colored in orange. The outflow is to the right.

According to the experiment,[2] at this shear rate the flow of the normal pooled plasma in a static capillary is occluded at $t \approx 180$ $[s]$.

We consider two scenarios of domain deformation in time, see Figure 14. The dumbbell-shaped deformation specific to the veins is

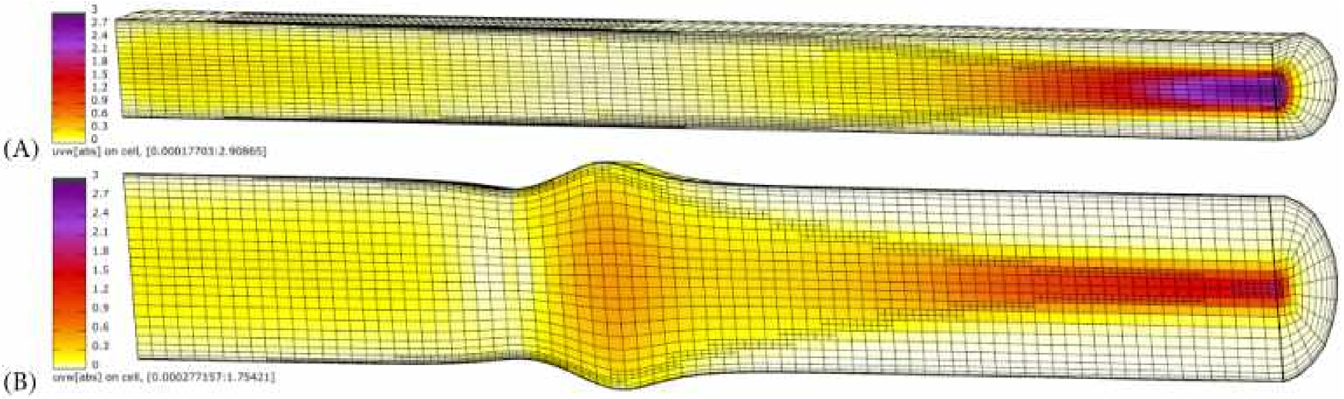
3
2.7
2.4
2.1
1.8
1.5
1.2
0.9
0.6
0.3
0
(A)
uvw[abs] on cell, [0.00017703:2.90865]
3
2.7
2.4
2.1
1.8
1.5
1.2
0.9
0.6
0.3
0
(B)
uvw[abs] on cell, [0.000277157:1.75421]

FIGURE 16 Middle cutaway of the mesh colored in velocity magnitude $|\mathbf{u}| \in [0,3]$ $(\mathrm{m\ s}^{-1})$ at $t = 87$ (s) for (A) dumbbell-shaped deformation and (B) pulse-wave deformation. The outflow is to the right.

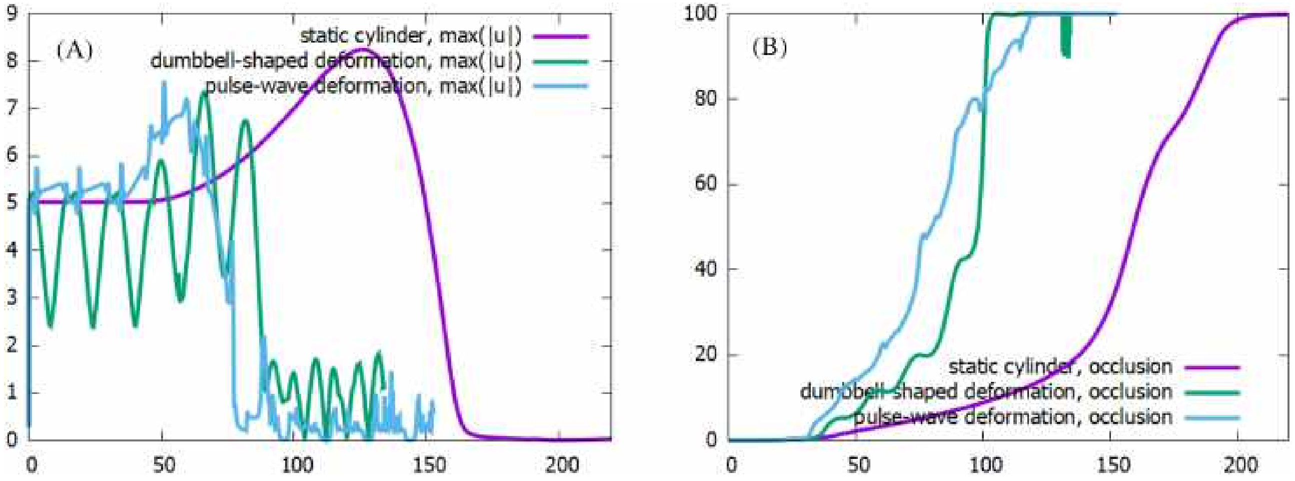

FIGURE 17 Maximum velocity $\max(|\mathbf{u}|)$ (A) and volumetric occlusion V_{acc} based on clot permeability K^{-1} (B).

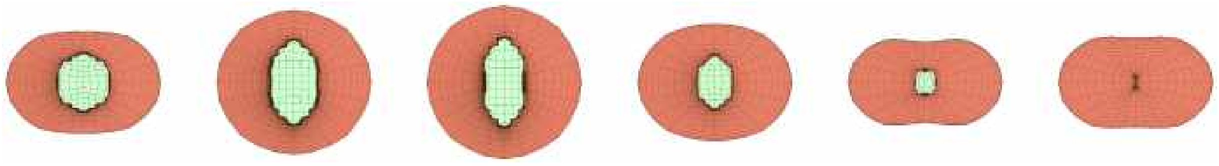

FIGURE 18 Front view of K^{-1}-isosurface 100 (mm^{-2}) at $t = \{75,78,81,84,87,90\}$ (s) for dumbbell-shaped deformation. The isosurface is colored in orange.

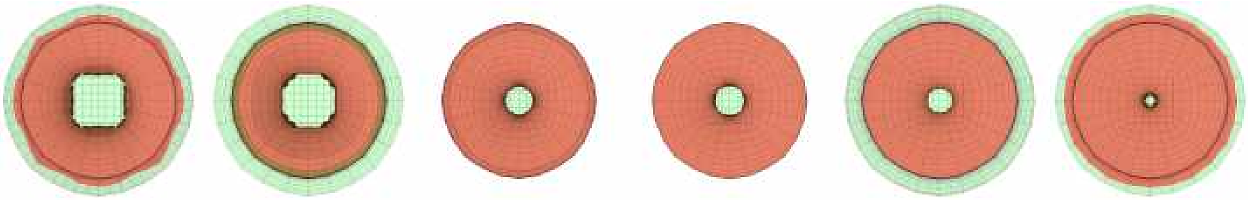

FIGURE 19 Front view of K^{-1}-isosurface 100 (mm^{-2}) at $t = \{72,75,78,81,84,87\}$ (s) for pulse-wave deformation. The isosurface is colored in orange.

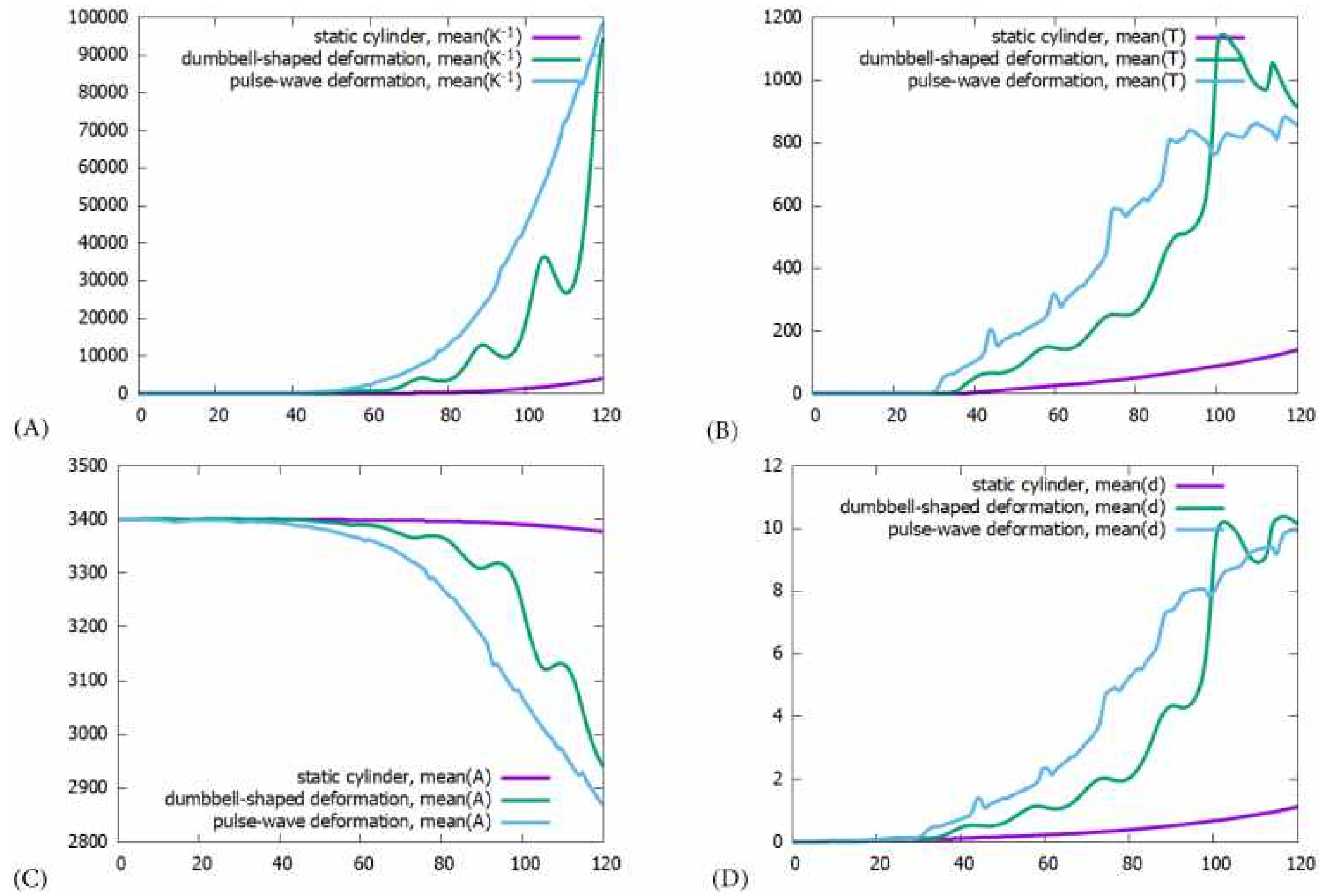

FIGURE 20 Mean value of (A) media resistivity K^{-1}, (B) thrombin concentration T, (C) antithrombin concentration A and (D) activated platelets count d over time interval $[0:120]$ (s).

$$\Omega_d = \left\{ \mathbf{x} = (x,y,z) : 0\ [mm] \leq z \leq L,\ \ x^2 + y^2 \leq \frac{R^2}{4}\left(3 + \cos\left(\frac{2\pi t}{16}\right) + \frac{x^2}{x^2+y^2}\left(1 - \cos\left(\frac{2\pi t}{16}\right)\right)\right)\right\}, \tag{96}$$

and the pulse-wave deformation is

$$\Omega_w = \left\{ \mathbf{x} = (x,y,z) : 0\ [mm] \leq z \leq L,\ \ x^2 + y^2 \leq \frac{R^2}{8}\left(8 + 2\operatorname{sech}\left(4\overline{z} - \frac{2}{5}\right) - \operatorname{sech}(4\overline{z} + 2)\right)\right\}, \tag{97}$$

where $\overline{z} = z + 2 - 12\left(\frac{t}{16} - \left\lfloor \frac{t}{16} \right\rfloor\right)$. In both cases the deformation is periodic, with a period of 16(s). In the case of dumbbell-shaped deformation, the cylinder is compressed from the top and bottom towards the center, which increases the pressure and reduces the inflow velocity. The pulse-wave expansion of the domain is followed by the contraction of the domain, propagates from the inlet to the outlet of the capillary and results in the decompression and compression of fluid in the expanded and contracted regions, respectively.

Three velocity components, pressure and 9 concentrations result in 13 unknowns per computational cell. The initial mesh consists of 20160 hexahedral cells, the dynamic refinement may increase the cells number till 50000. The nonlinear convergence criteria are $\tau_{\text{abs}} = 10^{-4}$, $\tau_{\text{rel}} = 10^{-2}$.

We refine the cells with media resistivity $K^{-1} \in [1,100]$ (mm^{-2}). These cells form the transition zone between the high and low permeability zones. In Figure 15 we show the K^{-1}-isosurface 100 (mm^{-2}) for all three types of capillaries at $t = 87$ (s). At $K^{-1} = 100$ (mm^{-2}) the fluid flow is obstructed by the clot. Indeed, Figure 16 shows that the velocity magnitude drops significantly in the region with $K^{-1} \geq 100($ mm$^{-2})$. Based on this, we evaluate the volumetric occlusion by

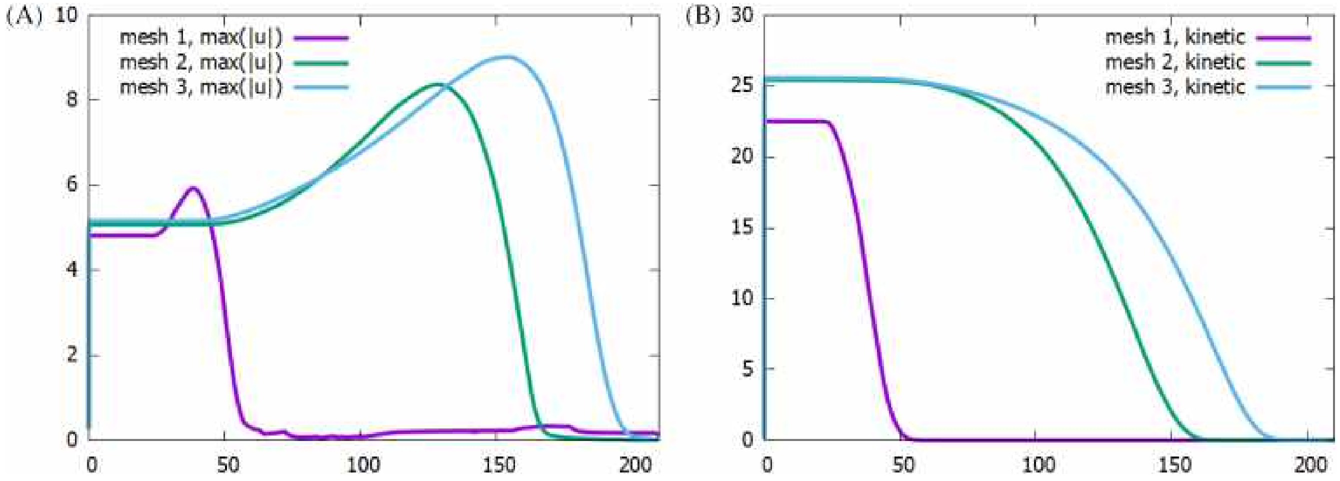

FIGURE 21 Maximum velocity $\max(|\mathbf{u}|)$ (A) and kinetic energy and (B) for a sequence of three meshes.

$$V_{occ} = \frac{1}{|\Omega|}\sum_{\omega_i} |\omega_i| \max\left(100^{-1}K^{-1}, 1\right) \cdot 100, \tag{98}$$

where $V_{\text{occ}} = 100$ (%) corresponds to the completely occluded channel.

The pulsating capillaries are occluded significantly earlier, at $t \approx 100$ (s), compared to the static capillary occlusion at $t \approx 200$ (s), see dynamics of the maximum velocity $\max(\|\mathbf{u}\|)$ and the volumetric occlusion V_{occ} in Figure 17. However, some fluid motion is still incurred by the domain deformation even after complete obstruction of the channel.

The evolution of the K^{-1}-isosurface is shown in Figures 18 and 19 for dumbbell-shaped and pulse-wave deformations, respectively.

Finally, in Figure 20 we demonstrate the evolution of the mean resistivity K^{-1} and mean concentrations of thrombin T, antithrombin A and activated platelets d.

The spikes in the velocity are observed in Figure 17 (left) in the case of pulse-wave shaped deformation due to narrowing of the inflow and outflow at certain moments in time.

In both cases deformation of the domain results in expulsion of some blood factors and partial declogging which results in some residual maximal velocity even with fully clogged cylinder, which is observed in Figure 17.

In some experiments with the dumbbell-shaped deformation we observed that the deformation-induced increase of pressure could reverse the flow at inflow and cause some stability issues.

In order to motivate our choice of the initial mesh with 20,160 hexahedral cells, we analyzed the maximum velocity and the density-normalized kinetic energy computed on 3 non-deforming and non-adapting quasiuniform meshes with 2760, 20160, 153600 hexahedral cells and maximum time step sizes 1/4, 1/4, 1/8 and presented in Figure 21. The coarsest mesh is inappropriate for this study, whereas the maximum velocity and the density-normalized kinetic energy computed on the medium-sized mesh demonstrate convergence.

11 | CONCLUSION

In this work, we considered the numerical method for the blood flow and coagulation modeling on dynamic adaptive moving meshes. We proposed a robust combination of moving mesh finite volume methods for the saddle-point system of Navier–Stokes equations, advection–diffusion equations for the blood factors, and coupled nonlinear advection–diffusion equations for the platelets. We combined these methods with the integration method for stiff reaction systems. These methods allow for the monolithic and fully implicit solution of the complete multiphysics system. The mesh adaptation allows us to increase accuracy in regions with flow peculiarities.

The future work will be directed towards fluid-porous structure interaction, models for capturing the blood rheology, and extension of the coagulation system for scenarios of clot formation without damaged endothelium.

ACKNOWLEDGMENTS

This work has been supported by the Russian Science Foundation, grant 21-71-20024 (numerical methods) and the Ministry of Education and Science of the Russian Federation, agreement 075-15-2022-286 (dynamic adaptive meshes). The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

CONFLICT OF INTEREST STATEMENT

The authors have no conflicts of interest to declare that are relevant to the content of this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Kirill M. Terekhov https://orcid.org/0000-0001-6548-0496
Ivan D. Butakov https://orcid.org/0000-0002-0424-6695
Alexander A. Danilov https://orcid.org/0000-0002-4709-4513
Yuri V. Vassilevski https://orcid.org/0000-0002-4718-1377

REFERENCES

1. Bouchnita A, Terekhov K, Nony P, Vassilevski Y, Volpert V. A mathematical model to quantify the effects of platelet count, shear rate, and injury size on the initiation of blood coagulation under venous flow conditions. *PLoS One*. 2020;15(7):e0235392.
2. Shen F, Kastrup CJ, Liu Y, Ismagilov RF. Threshold response of initiation of blood coagulation by tissue factor in patterned microfluidic capillaries is controlled by shear rate. *Arterioscler Thromb Vasc Biol*. 2008;28(11):2035-2041.
3. Terekhov KM. General finite-volume framework for saddle-point problems of various physics. *Russ J Numer Anal Math Model*. 2021;36(6):359-379.
4. Butakov ID, Terekhov KM. Two methods for the implicit integration of stiff reaction systems. *Comput Methods Appl Math*. 2022;23:83-92.
5. Bouchnita A. Mathematical modelling of blood coagulation and thrombus formation under flow in normal and pathological conditions. PhD thesis. Université Lyon 1-Claude Bernard; Ecole Mohammadia d'Ingénieurs-Université 2017.
6. Bouchnita A, Galochkina T, Kurbatova P, Nony P, Volpert V. Conditions of microvessel occlusion for blood coagulation in flow. *Int J Numer Methods Biomed Eng*. 2017;33(9):e2850.
7. Bouchnita A, Mozokhina A, Nony P, Llored JP, Volpert V. Combining computational modelling and machine learning to identify COVID-19 patients with a high thromboembolism risk. *Mathematics*. 2023;11(2):289.
8. Ladyzhenskaya O. *The Mathematical Theory of Viscous Incompressible Flow*. Gordon & Breach; 1969.
9. Babuška I. The finite element method with Lagrangian multipliers. *Numer Math*. 1973;20(3):179-192.
10. Brezzi F. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Publ Math Inform R*. 1974;S4:1-26.
11. Lebedev V. Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. I. *USSR Comput Math Math Phys*. 1964;4(3):69-92.
12. Harlow F, Welch J. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys Fluids*. 1965;8(12):2182-2189.
13. Olshanskii MA, Terekhov KM, Vassilevski YV. An octree-based solver for the incompressible Navier–Stokes equations with enhanced stability and low dissipation. *Comput Fluids*. 2013;84:231-246.
14. Perot B. Conservation properties of unstructured staggered mesh schemes. *J Comput Phys*. 2000;159(1):58-89.
15. Wenneker I, Segal G, Wesseling P. An unstructured staggered scheme for the Navier–Stokes equations. *Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2001 the 4th European Conference on Numerical Mathematics and Advanced Applications Ischia, July 2001*. Springer; 2003:189-197.
16. Rhie C, Chow W. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J*. 1983;21(11):1525-1532.
17. Brewster R, Carpenter C, Volpenhein E, Baglietto E, Smith J. Application of CD-adapco best practices to NESTOR OMEGA MVG Benchmark Exercises using STAR-CCM+. Proceedings of NURETH-16 2015.
18. ANSYS CFX-Solver Theory guide, release Ll. 2006.
19. Marković JĐ, Lukić NL, Ilić JD, Nikolovski BG, Sovilj MN, Šijački IM. Using the Ansys fluent for simulation of two-sided lid-driven flow in a staggered cavity. *Acta Period Technol*. 2012;43:169-178.
20. Rutkowski DR, Roldán-Alzate A, Johnson KM. Enhancement of cerebrovascular 4D flow MRI velocity fields using machine learning and computational fluid dynamics simulation data. *Sci Rep*. 2021;11(1):1-11.
21. Mangani L, Buchmayr M, Darwish M. Development of a novel fully coupled solver in openfoam: steady-state incompressible turbulent flows. *Numer Heat Transf Part B: Fundam*. 2014;66(1):1-20.

22. Mangani L, Buchmayr M, Darwish M, Moukalled F. A fully coupled OpenFOAM® solver for transient incompressible turbulent flows in ALE formulation. *Numer Heat Transf Part B: Fundam*. 2017;71(4):313-326.
23. Terekhov KM. Collocated finite-volume method for the incompressible Navier–Stokes problem. *J Numer Math*. 2021;29(1):63-79.
24. Terekhov KM. Fully-implicit collocated finite-volume method for the unsteady incompressible Navier–Stokes problem. *Numerical Geometry, Grid Generation and Scientific Computing*. Springer; 2021:361-374.
25. Terekhov KM. Presure boundary conditions in the collocated finite-volume method for the steady Navier–Stokes equations. *Comput Math Math Phys*. 2022;62(8):1343-1353.
26. Agélas L, Eymard R, Herbin R. A nine-point finite volume scheme for the simulation of diffusion in heterogeneous media. *C R Math*. 2009;347(11–12):673-676.
27. Vanleer B. *Flux-Vector Splitting for the 1990s*. In, NASA; 1991.
28. Glaister P. Flux difference splitting for hyperbolic systems of conservation laws with source terms. *Comput Math Appl*. 1993;26(7):79-96.
29. Hubbard ME, Garcia-Navarro P. Flux difference splitting and the balancing of source terms and flux gradients. *J Comput Phys*. 2000;165(1):89-125.
30. Oosterlee C, Ritzdorf H. Flux difference splitting for three-dimensional steady incompressible Navier–Stokes equations IN curvilinear co-ordinates. *Int J Numer Methods Fluids*. 1996;23(4):347-366.
31. Parent B. Positivity-preserving flux difference splitting schemes. *J Comput Phys*. 2013;243:194-209.
32. Parent B. Multidimensional flux difference splitting schemes. *AIAA J*. 2015;53(7):1936-1948.
33. Kuzmin D. Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes. *J Comput Appl Math*. 2012;236(9):2317-2337.
34. Terekhov K, Vassilevski Y. Finite volume method for coupled subsurface flow problems, I: Darcy problem. *J Comput Phys*. 2019;395:298-306.
35. Terekhov K, Tchelepi H. Cell-centered finite-volume method for elastic deformation of heterogeneous media with fulltensor properties. *J Comput Appl Math*. 2020;364:112331.
36. Tripuraneni SRT, Novikov A, Voskov D. *Non-linear finite volume discretization for subsurface flow and mechanics problem*. MSc thesis, Delft University of Technology, Netherlands; 2021.
37. Terekhov KM. Multi-physics flux coupling for hydraulic fracturing modelling within INMOST platform. *Russ J Numer Anal Math Model*. 2020;35(4):223-237.
38. Terekhov K. Cell-centered finite-volume method for heterogeneous anisotropic poromechanics problem. *J Comput Appl Math*. 2020;365(112357):112357.
39. Novikov A, Voskov D, Khait M, Hajibeygi H, Jansen J. A collocated finite volume scheme for high-performance simulation of induced seismicity in geo-energy applications. *SPE Reservoir Simulation Conference*. OnePetro; 2021.
40. Novikov A, Voskov D, Khait M, Hajibeygi H, Jansen JD. A scalable collocated finite volume scheme for simulation of induced fault slip. *J Comput Phys*. 2022;469:111598.
41. Terekhov KM, Vassilevski YV. Finite volume method for coupled subsurface flow problems, II: Poroelasticity. *J Comput Phys*. 2022;462:111225.
42. Gresho PM, Sani RL. On pressure boundary conditions for the incompressible Navier–Stokes equations. *Int J Numer Methods Fluids*. 1987;7(10):1111-1145.
43. Gresho P. On pressure boundary conditions for the incompressible Navier-Stokes equations. *Int J Numer Methods Fluids*. 1990;7:11-46.
44. Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. *Comput Methods Appl Mech Eng*. 2006;195(44–47):6011-6045.
45. Sani RL, Shen J, Pironneau O, Gresho P. Pressure boundary condition for the time-dependent incompressible Navier–Stokes equations. *Int J Numer Methods Fluids*. 2006;50(6):673-682.
46. Braack M, Mucha PB. Directional do-nothing condition for the Navier-Stokes equations. *J Comput Math*. 2014;32:507-521.
47. Tuković Ž, Jasak H. A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. *Comput Fluids*. 2012;55:70-84.
48. Perot B, Nallapati R. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *J Comput Phys*. 2003;184(1):192-214.
49. Perot JB, Schmidt DP. Unstructured adaptive moving mesh solution of unsteady shear flows and free-surface flows. *Proceedings of the Fourth International Symposium on Turbulence and Shear Flow Phenomena*. Begel House Inc.; 2005.
50. Jasak H, Tukovic Z. Automatic mesh motion for the unstructured finite volume method. *Trans Famena*. 2006;30(2):1-20.
51. Batina JT. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA J*. 1990;28(8):1381-1388.
52. Baker T. Mesh modification for solution adaptation and time evolving domains. Proceedings of the 7th international conference on numerical grid generation in computational field simulations. 2000.
53. Huang W, Ren Y, Russell RD. Moving mesh methods based on moving mesh partial differential equations. *J Comput Phys*. 1994;113(2):279-290.
54. Huang W, Russell RD. Adaptive mesh movement—the MMPDE approach and its applications. *J Comput Appl Math*. 2001;128(1–2):383-398.

55. Zwart P, Raithby G, Raw M. The integrated space-time finite volume method and its application to moving boundary problems. *J Comput Phys*. 1999;154(2):497-519.
56. Zwart PJ. *The Integrated Space-Time Finite Volume Method*. PhD thesis, University of Waterloo, Canada; 1999.
57. Rendall TC, Allen CB, Power ED. Conservative unsteady aerodynamic simulation of arbitrary boundary motion using structured and unstructured meshes in time. *Int J Numer Methods Fluids*. 2012;70(12):1518-1542.
58. Florian T. Space-time finite volume method in OpenFOAM. PhD thesis. Wien, 2023.
59. Takizawa K. Computational engineering analysis with the new-generation space–time methods. *Comput Mech*. 2014;54:193-211.
60. Terahara T, Takizawa K, Tezduyar TE, Bazilevs Y, Hsu MC. Heart valve isogeometric sequentially-coupled FSI analysis with the space–time topology change method. *Comput Mech*. 2020;65:1167-1187.
61. Terahara T, Takizawa K, Tezduyar TE, Tsushima A, Shiozaki K. Ventricle-valve-aorta flow analysis with the space–time isogeometric discretization and topology change. *Comput Mech*. 2020;65:1343-1363.
62. Kucharik M, Shashkov M. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2014;258:268-304.
63. Margolin L, Shashkov M. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J Comput Phys*. 2003; 184(1):266-298.
64. Sadrehaghighi I. Dynamic and adaptive meshing. CFD Open Series, Patch 2.23, 2021.
65. Ceniceros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. *J Comput Phys*. 2001;172(2):609-639.
66. Tu T, O'Hallaron DR, Ghattas O. Scalable parallel octree meshing for terascale applications. *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. IEEE; 2005:4.
67. Terekhov K, Vassilevski Y. Two-phase water flooding simulations on dynamic adaptive octree grids with two-point nonlinear fluxes. *Russ J Numer Anal Math Model*. 2013;28(3):267-288.
68. Terekhov KM, Nikitin KD, Olshanskii MA, Vassilevski YV. A semi-Lagrangian method on dynamically adapted octree meshes. *Russ J Numer Anal Math Model*. 2015;30(6):363-380.
69. George P, Hecht F. Nonisotropic grids. *Nonisotropic Grids*. CRC Press; 1999:20-21.
70. Gorshkova E, Neittaanmäki P, Repin S. Comparative study of the a posteriori error estimators for the Stokes problem. *Numerical Mathematics and Advanced Applications: Proceedings of the 6th European Conference on Numerical Mathematics and Advanced Applications Santiago de Compostela, Spain, July 2005*. Springer; 2006:252-259.
71. Gorshkova E, Neittaanmaki P, Repin S. Mesh-adaptive methods for viscous flow problem with rotation. *Advances and Innovations in Systems, Computing Sciences and Software Engineering*. Springer; 2007:105-107.
72. Boyer F, Lapuerta C, Minjeaud S, Piar B, Quintard M. Cahn–Hilliard/Navier–Stokes model for the simulation of threephase flows. *Transp Porous Media*. 2010;82:463-483.
73. Holmes D, Connell S. Solution of the 2D Navier-Stokes equations on unstructured adaptive grids. *Proceedings of the 9th Computational Fluid Dynamics Conference*. Vol 1989. AIAA; 1932.
74. Wang Z. A quadtree-based adaptive Cartesian/quad grid flow solver for Navier–Stokes equations. *Comput Fluids*. 1998;27(4):529-549.
75. Nemec M, Aftosmis M, Wintzer M. Adjoint-based adaptive mesh refinement for complex geometries. *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA; 2008:725.
76. Roy C. Strategies for driving mesh adaptation in CFD. *Proceedings of the 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. AIAA; 2009:1302.
77. Losasso F, Gibou F, Fedkiw R. Simulating water and smoke with an octree data structure. *ACM Siggraph 2004 Papers*. ACM; 2004: 457-462.
78. Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J Comput Phys*. 2003; 190(2):572-600.
79. Van Hooft JA, Popinet S, Van De Wiel BJ. Adaptive cartesian meshes for atmospheric single-column models: a study using basilisk 18-02-16. *Geosci Model Dev*. 2018;11(12):4727-4738.
80. Terekhov K. Parallel dynamic mesh adaptation within INMOST platform. *Russian Supercomputing Days*. Springer; 2019:313-326.
81. Vassilevski YV, Terekhov K. Nonlinear finite volume method for the Interface advection-compression problem on unstructured adaptive meshes. *Comput Math Math Phys*. 2022;62(7):1041-1058.
82. Batty C. A cell-centred finite volume method for the Poisson problem on non-graded quadtrees with second order accurate gradients. *J Comput Phys*. 2017;331:49-72.
83. Auzinger W, Frank R, Kirlinger G. Modern convergence theory for stiff initial-value problems. *J Comput Appl Math*. 1993;45(1–2):5-16.
84. Dahlquist GG. A special stability problem for linear multistep methods. *BIT Numer Math*. 1963;3(1):27-43.
85. Dahlquist G. *On Stability and Error Analysis for Stiff Non-linear Problems Part I. Tecnical Report CM-P00069396*. The Royal Institute of Technology; 1975.
86. Liu M, Zhang L, Zhang C. Study on banded implicit Runge–Kutta methods for solving stiff differential equations. *Math Probl Eng*. 2019;2019:1-8.
87. Auzinger W, Frank R. Asymptotic error expansions for stiff equations: an analysis for the implicit midpoint and trapezoidal rules in the strongly stiff case. *Numer Math*. 1989;56(5):469-499.
88. Vassilevski Y, Konshin I, Kopytov G, Terekhov K. *INMOST—A Software Platform and a Graphical Environment for Development of Parallel Numerical Models on General Meshes*. Moscow State University Publications; 2013.

89. Vassilevski Y, Terekhov K, Nikitin K, Kapyrin I. *Parallel Finite Volume Computation on General Meshes*. Springer; 2020.
90. Danilov AA, Terekhov KM, Konshin IN, Vassilevski YV. Parallel software platform INMOST: a framework for numerical modeling. *Supercomput Front Innov*. 2015;2(4):55-66.
91. Terekhov K, Vassilevski Y. Mesh modification and adaptation within INMOST programming platform. *Numerical Geometry, Grid Generation and Scientific Computing*. Springer; 2019:243-255.
92. Bastian P, Blatt M, Dedner A, et al. The Dune framework: basic concepts and recent developments. *Comput Math Appl*. 2021;81:75-112.
93. Flemisch B, Darcis M, Erbertseder K, et al. DuMux: DUNE for multi-phase, component, scale, physics, flow and transport in porous media. *Adv Water Resour*. 2011;34(9):1102-1112.
94. Koch T, Gläser D, Weishaupt K, et al. DuMux 3–an open-source simulator for solving flow and transport problems in porous media with a focus on model coupling. *Comput Math Appl*. 2021;81:423-443.
95. The Trilinos Project Team. The Trilinos project website. Accessed November 5, 2022. https://trilinos.github.io 2022.
96. Rettenmaier D, Deising D, Ouedraogo Y, et al. Load balanced 2D and 3D adaptive mesh refinement in OpenFOAM. *SoftwareX*. 2019; 10:100317.
97. Tautges TJ, Ernst C, Stimpson C, Meyers RJ, Merkley K. *MOAB: a mesh-oriented database. Technical Report*. Sandia National Laboratories; 2004.
98. Garimella RV. *MSTK-A flexible infrastructure library for developing mesh based applications*. IMR; 2004:213-220.
99. Vassilevski Y, Danilov A, Lozovskiy A, et al. A stable method for 4D CT-based CFD simulation in the right ventricle of a TGA patient. *Russ J Numer Anal Math Model*. 2020;35(5):315-324.
100. Terekhov K. INMOST—a toolkit for distributed mathematical. Accessed November 5, 2022 modelling. http://inmost.org/ 2022.
101. Rand RH. *Introduction to Maxima*. Vol 25. Cornel University; 2005:22.
102. Haynsworth EV. Determination of the inertia of a partitioned Hermitian matrix. *Linear Algebra Appl*. 1968;1(1):73-81.
103. Terekhov K. Parallel multilevel linear solver within INMOST platform. *Russian Supercomputing Days*. Springer; 2020:297-309.
104. Terekhov K. *Greedy dissection method for shared parallelism in incomplete factorization within INMOST platform, Russian Supercomputing Days*. Springer; 2021:87-101.
105. Brandt A, Livne OE. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics. Revised Edition*. SIAM; 2011.
106. Ethier CR, Steinman D. Exact fully 3D Navier–Stokes solutions for benchmarking. *Int J Numer Methods Fluids*. 1994;19(5):369-375.
107. Lozovskiy A, Olshanskii MA, Vassilevski YV. A quasi-Lagrangian finite element method for the Navier–Stokes equations in a time-dependent domain. *Comput Methods Appl Mech Eng*. 2018;333:55-73.
108. Danilov A, Lozovskiy A, Olshanskii M, Vassilevski Y. A finite element method for the Navier-Stokes equations in moving domain with application to hemodynamics of the left ventricle. *Russ J Numer Anal Math Model*. 2017;32(4):225-236.
109. Yushkevich PA, Piven J, Hazlett HC, et al. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *NeuroImage*. 2006;31(3):1116-1128.
110. Rineau L, Yvinec M. Ageneric software design for Delaunay refinement meshing. *Comput Geom*. 2007;38(1–2):100-110.
111. Lipnikov K, Vassilevski Y, Danilov A. Advanced numerical instruments 3D. Accessed November 5, 2022. 2017 https://sourceforge.net/projects/ani3d
112. Sadovnichy V, Tikhonravov A, Voevodin V, Opanasenko V. "Lomonosov": supercomputing at Moscow state university. *Contemporary High Performance Computing*. Chapman and Hall/CRC; 2017:283-307.
113. Voevodin VV, Antonov AS, Nikitenko DA, et al. Supercomputer Lomonosov-2: large scale, deep monitoring and fine analytics for the user community. *Supercomput Front Innov*. 2019;6(2):4-11.
114. Leiderman K, Fogelson AL. Grow with the flow: a spatial–temporal model of platelet deposition and blood coagulation under flow. *Math Med Biol*. 2011;28(1):47-84.
115. Wufsus AR, Macera N, Neeves K. The hydraulic permeability of blood clots as a function of fibrin and platelet density. *Biophys J*. 2013; 104(8):1812-1823.
116. Hockin MF, Jones KC, Everse SJ, Mann KG. A model for the stoichiometric regulation of blood coagulation. *J Biol Chem*. 2002;277(21): 18322-18333.
117. Wiebe EM, Stafford AR, Fredenburgh JC, Weitz JI. Mechanism of catalysis of inhibition of factor IXa by antithrombin in the presence of heparin or pentasaccharide. *J Biol Chem*. 2003;278(37):35767-35774.
118. Galochkina T, Bouchnita A, Kurbatova P, Volpert V. Reaction-diffusion waves of blood coagulation. *Math Biosci*. 2017;288:130-139.
119. Tsiang M, Paborsky L, Li WX, et al. Protein engineering thrombin for optimal specificity and potency of anticoagulant activity in vivo. *Biochemistry*. 1996;35(51):16449-16457.
120. Kuharsky AL, Fogelson AL. Surface-mediated control of blood coagulation: the role of binding site densities and platelet deposition. *Biophys J*. 2001;80(3):1050-1074.

How to cite this article: Terekhov KM, Butakov ID, Danilov AA, Vassilevski YV. Dynamic adaptive moving mesh finite-volume method for the blood flow and coagulation modeling. *Int J Numer Meth Biomed Engng*. 2023; e3731. doi:10.1002/cnm.3731