Radka Keslerová; Anna Lancmanová; Tomáš Bodnár

Validation of numerical simulations of a simple immersed boundary solver for fluid flow in branching channels

In: Jan Chleboun and Pavel Kůs and Jan Papež and Miroslav Rozložník and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Jablonec nad Nisou, June 19-24, 2022. Institute of Mathematics CAS, Prague, 2023. pp. 85–96.

Persistent URL: http://dml.cz/dmlcz/703191

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

VALIDATION OF NUMERICAL SIMULATIONS OF A SIMPLE IMMERSED BOUNDARY SOLVER FOR FLUID FLOW IN BRANCHING CHANNELS

Radka Keslerová[1], Anna Lancmanová[1,2], Tomáš Bodnár[1,2]

[1]Czech Technical University in Prague, Faculty of Mechanical Engineering,
Department of Technical Mathematics,
Karlovo nám. 13, Prague, 121 35, Czech Republic
[2]Institute of Mathematics, Czech Academy of Sciences,
Žitná 25, 115 67 Prague 1, Czech Republic.
Radka.Keslerova@fs.cvut.cz, Anna.Lancmanova@fs.cvut.cz,
Tomas.Bodnar@fs.cvut.cz

Abstract: This work deals with the flow of incompressible viscous fluids in a two-dimensional branching channel. Using the immersed boundary method, a new finite difference solver was developed to interpret the channel geometry. The numerical results obtained by this new solver are compared with the numerical simulations of the older finite volume method code and with the results obtained with OpenFOAM. The aim of this work is to verify whether the immersed boundary method is suitable for fluid flow in channels with more complex geometries with difficult grid generation.

Keywords: immersed boundary method, finite volume method, OpenFOAM

MSC: 65L06, 65N08, 76A05, 76A10, 76D05

1. Introduction

Fluid flow in the system of branching channels is a part of many technical or biological applications, for example blood flow in the fine and complex branching of the cardiovascular system. This work is focused on the flow of blood in the venous system, for simplification it is possible to consider blood flow as flow of incompressible viscous fluid in branching channels.

The network of such a branching system can be imagined as a main channel followed by multilevel branching, and each of these new branches can have a different diameter and can be connected to the main channel at a different angle.

Complex formation of the channel system causes problems related to the description of the geometry, its mesh generation, and the mathematical formulation of the related problem, including appropriate boundary conditions. The description of the

DOI: 10.21136/panm.2022.09

channel geometry can be done using the standard grid generation inside the channel. A grid can be either structured or unstructured. This approach is quite common, but it is associated with certain disadvantages. This includes the difficulty of mesh generation and the need to re-generate the mesh in case of even small geometric modifications. Also, CFD solvers for general unstructured grids are more complex, making it difficult to implement any non-standard mathematical models or boundary conditions.

Some problems that arise when using classical methods on grids inside the area (limited by the channel boundary) can be avoided by adopting the immersed boundary method. In this case a larger area of space is discretized, e.g. the rectangle enclosing the tested branch channel. A grid (Cartesian grid) is constructed throughout such a domain, where model equations are also solved. The specific geometry of the channel is represented only at the level of the mathematical model used, one model in the region occupied by the fluid and another elsewhere. Switching between models is simply implemented using a characteristic function specifying the inner and outer parts of the considered channel. In this case, due to the very simple grid structure and domain shape, the CFD solver can be very simple. Any changes in the geometry of the channel are easily solved, it is only necessary to redefine the characteristic function describing the fluid region.

The aim of this work is to compare the results of a standard method based on finite volumes, which uses the grid built inside the channel, i.e. the grid bounded by the channel edges, with a much simpler finite difference code working on the regular Cartesian grid using a general implementation of the immersed boundary method. A simple straight channel with one branch inclined at different angles was chosen as the test case.

2. Mathematical model

The fundamental system of equations is the system of Navier–Stokes equations for incompressible Newtonian fluids. This system is based on the balance laws of mass and momentum for incompressible fluids

$$\operatorname{div} \boldsymbol{u} = 0 \tag{1}$$

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{u}) \right) = -\nabla P + \mu \Delta \boldsymbol{u}, \tag{2}$$

where P is the pressure, ρ is the constant density, $\boldsymbol{u}$ is the velocity vector and μ represents the constant dynamic viscosity.

3. Numerical methods

The numerical methods which solve the system of incompressible Navier-Stokes equations can be divided according to velocity-pressure coupling stategy into two main groups, coupled methods (e.g. artificial compressibility and dual time-stepping

methods) and pressure correction methods (including e.g. SIMPLE or PISO algorithms). The SIMPLE algorithm [11, 12] is the main method used for the numerical solution of incompressible fluid flow problems (also due to its ability to treat unsteady flows). This algorithm is included in Open source Field Operation And Manipulation (OpenFOAM) and is described in detail in [7, 14].

The artificial compressibility method (designed to treat steady flows) was used in our in-house built FDM and FVM codes. This method [4, 6] is used to obtain equation for pressure. It means that the continuity equation is completed by a pressure time derivative term $\frac{\partial p}{\beta^2 \partial t}$, where β is a positive parameter, making the inviscid part of the system of equations hyperbolic. The parameter β for the steady case is chosen approximately equal to the maximum velocity in the domain.

3.1. Finite difference method

The finite difference approximation of the governing system of equations (1) and (2) is a natural choice because of the use of immersed boundary method on Cartesian grids. In such case the discretization is simple, allowing for easy implementation and modification of various numerical methods and algorithms.

The system including the modified (for artificial compressibility) continuity equation (1) and the momentum equation (2) can be written in vector form as [1]:

$$\boldsymbol{D}_\beta \mathbf{W}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{R}_x + \mathbf{S}_y, \tag{3}$$

where $\boldsymbol{D}_\beta = \operatorname{diag}\left(\frac{1}{\rho\beta^2}, 1, 1\right)$, $\mathbf{W} = \operatorname{col}(p, u, v)$ is the vector of unknowns,

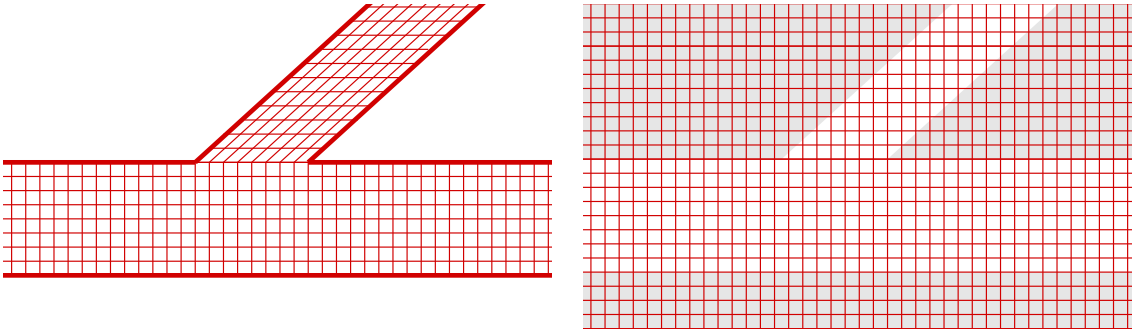
$$\mathbf{F} = \begin{pmatrix} u \\ u^2 + p \\ vu \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} v \\ uv \\ v^2 + p \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0 \\ \nu u_x \\ \nu v_x \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \nu u_y \\ \nu v_y \end{pmatrix} \tag{4}$$

where p is the kinematic pressure ($p = P/\rho$), u, v are velocity components and ν is the kinematic viscosity.

3.1.1. Immersed boundary method

In computational fluid dynamics, the immersed boundary method was first used in reference to the method developed by Charles Peskin in 1972 (see [13]) to simulate fluid-structure interactions.

A characteristic feature of this method is that the numerical simulation of fluid flow is performed on Cartesian grid that does not have to directly copy the geometry of the computational (fluid) domain, see e.g. [3, 10]. The situation can be described using the schematic sketch (shown in Fig. 1) of the grids used for Finite Volume Method (FVM) and Finite Difference Method (FDM) in this work. The structured grid used in FVM simulations has a simple structure with the grid lines fitted to boundaries of the computational domain. This results in grids that are aligned to boundaries. This situation is shown in Fig. 1 (a).



(a) Grid for FVM (b) Grid for FDM with the immersed geometry

Figure 1: Detail of the grid for finite-volume and finite-difference simulations.

In the immersed boundary FDM method, the governing system of equations is discretized in the whole rectangular domain and used boundary conditions are only imposed on its boundary. The unknown values of velocity and pressure are sought in all internal points of the domain, distinguishing the points inside of the fluid domain (marked by white color in Fig. 1 (b)) and inside of the solid domain (marked by gray color in Fig. 1 (b)). The velocity fields in the solid domain is set to zero, so that the governing equations are only solved in the points in the fluid region, see e.g. [3, 8].

3.1.2. MacCormack scheme

In computational fluid dynamics, the MacCormack method is a widely used discretization method for the numerical solution of hyperbolic partial differential equations. This second-order finite difference method was introduced by Robert W. MacCormack in 1969 [9]. It is the method written in the predictor-corrector form using asymmetric forward/backward discretization stencil to approximate spatial derivatives to provide finally a central (second order) approximation.

To describe the MacCormack scheme, rearrange equation (3) to the form where all terms except the time derivative are placed on the right hand side

$$\mathbf{W}_t = \boldsymbol{D}_\beta^{-1}\left[-\left(\mathbf{F}_x + \mathbf{G}_y\right) + \nu \boldsymbol{D}\Delta\mathbf{W}\right], \quad \nu\boldsymbol{D}\Delta\mathbf{W} = \mathbf{R}_x + \mathbf{S}_y. \tag{5}$$

To update in time the values of the vector $\mathbf{W}^n$ to $\mathbf{W}^{n+1}$ an approximation of $\mathbf{W}_t$ is constructed from (5). This approximation is built differently, asymmetrically, in predictor (e.g. with backward differences) and in corrector (by forward differences). The final update is performed using linear combination of the two values obtained. The expressions for predicted and corrected values are shown in (6) and (7).

$$\begin{aligned}\widetilde{\mathbf{W}}_{i,j} =& \mathbf{W}_{i,j}^n + \Delta t\, \boldsymbol{D}_\beta^{-1}\Bigg[- \frac{\mathbf{F}_{i,j}^n - \mathbf{F}_{i-1,j}^n}{\Delta x} - \frac{\mathbf{G}_{i,j}^n - \mathbf{G}_{i,j-1}^n}{\Delta y} \\ &+ \nu\, \boldsymbol{D}\left(\frac{\mathbf{W}_{i+1,j}^n - 2\mathbf{W}_{i,j}^n + \mathbf{W}_{i-1,j}^n}{\Delta x^2} + \frac{\mathbf{W}_{i,j+1}^n - 2\mathbf{W}_{i,j}^n + \mathbf{W}_{i,j-1}^n}{\Delta y^2}\right)\Bigg]\end{aligned} \tag{6}$$

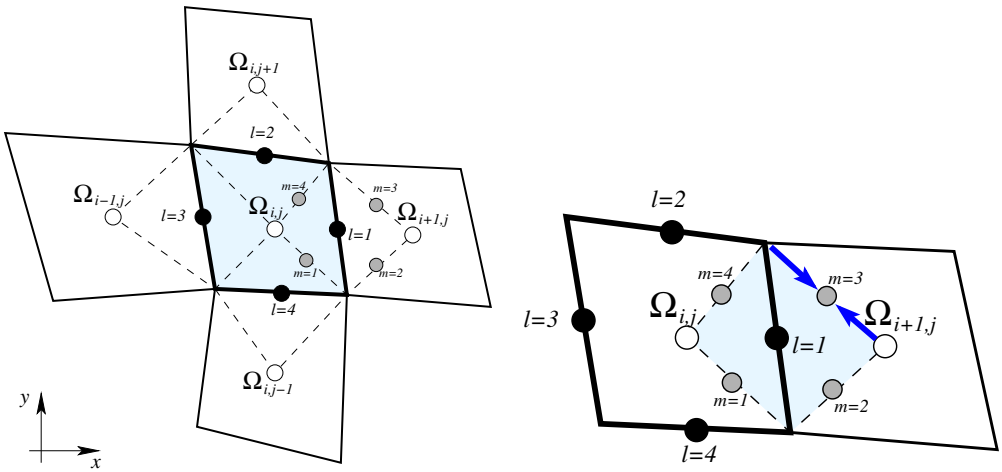
$$\mathbf{W}_{i,j}^{n+1} = \frac{1}{2}\left(\mathbf{W}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i,j}\right) + \frac{\Delta t}{2}\,\boldsymbol{D}_{\beta}^{-1}\Bigg[- \frac{\widetilde{\mathbf{F}}_{i+1,j}^{n} - \widetilde{\mathbf{F}}_{i,j}^{n}}{\Delta x} - \frac{\widetilde{\mathbf{G}}_{i,j+1}^{n} - \widetilde{\mathbf{G}}_{i,j}^{n}}{\Delta y} \tag{7}$$

$$+ \nu\,\boldsymbol{D}\left(\frac{\widetilde{\mathbf{W}}_{i+1,j}^{n} - 2\widetilde{\mathbf{W}}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i-1,j}^{n}}{\Delta x^2} + \frac{\widetilde{\mathbf{W}}_{i,j+1}^{n} - 2\widetilde{\mathbf{W}}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i,j-1}^{n}}{\Delta y^2}\right)\Bigg]. \tag{8}$$

3.2. Finite volume method

In this work the finite volume discretization is used as a reference for comparison and validation of the newly developed finite-difference solver. Within the presented study, the finite volume method was used in two codes. First, in an in-house developed simple 2D code, and second, in OpenFOAM.

The spatial discretization is based on the cell-centered finite volume approximation on a multi-block structured grid. While the mesh is handled as block structured by the in-house solver, the same grid is treated as unstructured by OpenFOAM. The finite volumes are quadrilaterals in 2D. For the in-house code the central scheme is used for convective terms, including the pressure gradient calculated from the approximation. The viscous terms are also discretized in the central way on dual quadrilateral mesh (diamond type scheme), see Fig. 2.



(a) Primary & secondary cells (b) Approximation of viscous fluxes

Figure 2: Grid configuration for approximation of inviscid and viscous fluxes.

The resulting semi-discrete system of ODEs (based on (5)) is integrated in time by the explicit multistage Runge–Kutta scheme:

$$\begin{aligned} \mathbf{W}_{i,j}^{(0)} &= \mathbf{W}_{i,j}^{n} \\ \mathbf{W}_{i,j}^{(r+1)} &= \mathbf{W}_{i,j}^{(0)} - \alpha_{(r)}\Delta t\mathcal{L}\mathbf{W}_{i,j}^{(r)} \qquad r = 1,\ldots,s \\ \mathbf{W}_{i,j}^{n+1} &= \mathbf{W}_{i,j}^{(s)} \end{aligned} \tag{9}$$

The three-stage explicit Runge-Kutta scheme used to obtain results presented here had coefficients: $\alpha_{(1)} = 1/2$, $\alpha_{(2)} = 1/2$, $\alpha_{(3)} = 1$. More details on this type of finite volume discretization and associated Runge-Kutta methods can be found, e.g., in [1, 2, 5].

OpenFOAM uses a co-located grid, i.e., the fluid dynamic quantities are all stored at the control volumes centroids. The convective terms are discretized using the central difference scheme and also for the viscous fluxes the central differences are used. In this case, however, due to grid curvature an extra correction term (for non-orthogonality) is added to the discretization, subject to certain limiter, for more details see [11].

4. Numerical tests

The numerical results shown in this section are used to compare different numerical methods to verify that the newly developed immersed boundary method is sufficiently accurate. At the same time, the flow at the branching point of the channel was also tested depending on the connection angle of the secondary branch.

4.1. Domain geometry

For the immersed boundary implementation of finite-difference method, the $2D$ computational domain was chosen as a rectangle in $x - y$ plane with dimensions $30D \times 10D$. The numerical simulations were performed on the structured (Cartesian) grid with different number of equidistant nodes.

The used domain is shown in Fig. 3. The diameter of the main channel is denoted by symbol D and $D = 0.006\ m$ and the width of the branch inclined at the angle α was chosen to be $D/2$. The same configuration was kept for all simulations, just changing the angle α by setting it to values 30°, 60°, 90°, 120° and 150°. For finite volume simulations, just the interior of the channel (marked by white color in Fig. 3) was used to construct the grid.

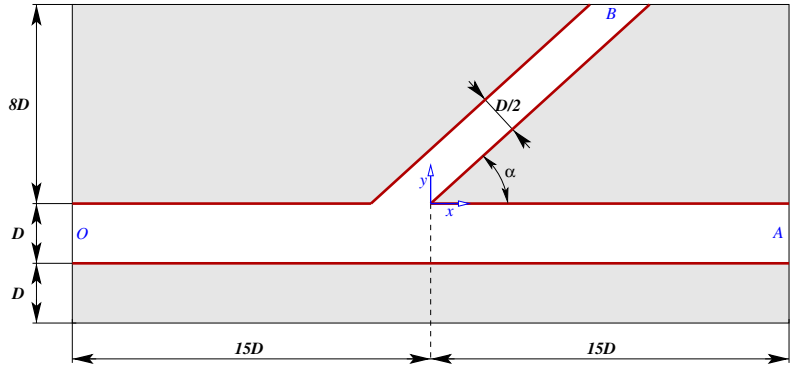


Figure 3: Computational domain of a planar branching channel.

4.2. Boundary conditions

Boundary conditions were chosen in such a way, that the flow is driven by the prescribed pressure drop between the inlet and outlet parts of the boundary. So only different values of pressure were prescribed at inlet ($p_{in} = 60\ Pa$) and in outlet parts ($p_A = p_B = 0\ Pa$) of the boundary. Otherwise the homogeneous Neumann condition was prescribed for velocity components on those parts of boundary to mimic a fully developed flow. On the channel wall of course the no-slip, i.e., homogeneous Dirichlet condition $\boldsymbol{u} = (0, 0)$ was prescribed for velocity.

4.3. Numerical results

The aim of presented numerical results is to demonstrate the applicability of the chosen methods and their settings for the considered class of problems. The newly developed FD method based immersed boundary code is compared with an in-house finite volume code (both using artificial compressibility approach) and the open-source OpenFOAM finite volume code (using a variant of SIMPLE algorithm). Both FVM codes share the same computational grid. For the FDM method with immersed boundary channel representation two different grids were used. The coarser grid had resolution 1200×200 cells, while the finer grid doubled the number of cells in the vertical y direction, i.e., having 1200×400 cells.

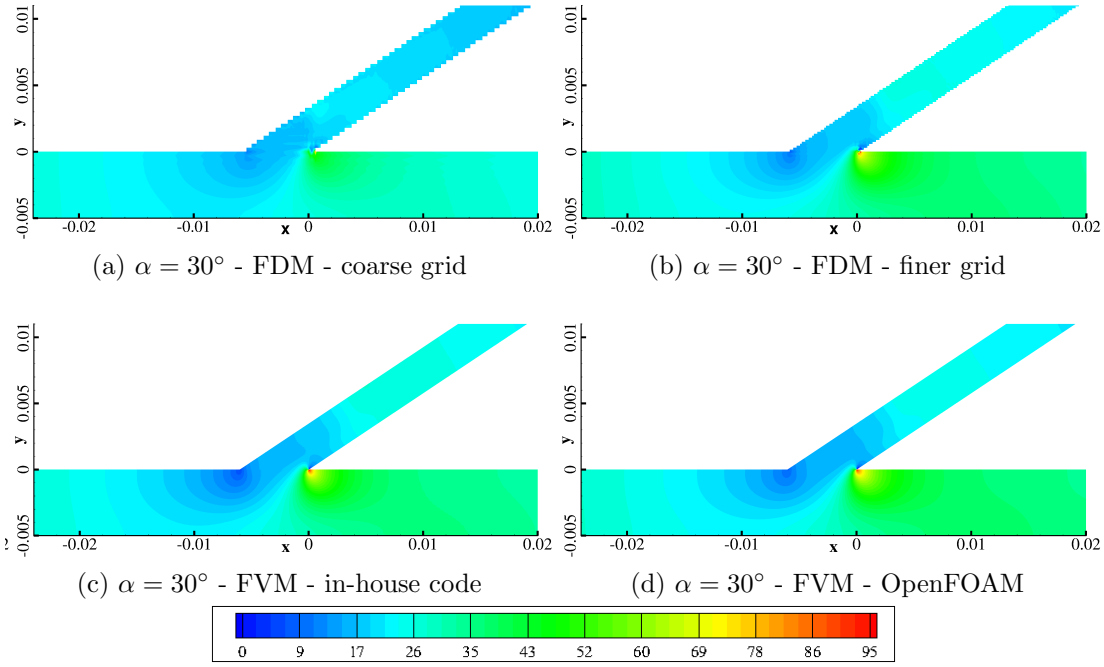
Figs. 4 and 5 show the comparison of pressure and velocity fields obtained using all the considered codes for the case of oblique branching at angle $\alpha = 30°$. The pressure fields in Fig. 4 have very similar character and except the FDM results on coarse grid all results are almost identical. The comparison of velocity magnitude fields in Fig. 5 reveals that the in-house FVM code and FDM code on the finer grid provide almost identical results. The OpenFOAM results predict a bit higher velocity in the main channel, while the FDM code on coarse grid predicts lower velocity.

It is interesting to see that the level of agreement between the results changes for different angles α of the secondary branch. The comparison of pressure and velocity fields in the case of $\alpha = 60°$ is shown in Figs. 6 and 7. Here it seems that the OpenFOAM results are closest to the FDM on the finer grid.

The comparison of results in the case of $\alpha = 90°$ (shown in Figs. 8 and 9) shows that even the results obtained by FDM on the coarse grid are almost identical to the other methods. The orthogonality of the grid allows for optimal use of all computational points and leads to highest accuracy of numerical approximation.

Similar results were obtained for the remaining two tested angles, $\alpha = 120°$ and $\alpha = 150°$ (not shown here). Also here the mutual agreement between the finer grid of immersed boundary method and the in-house code can be seen.

Similar comparison for cases with different branching angle α is shown in presented Figs. 4-9 for finite volume method (in-house code and OpenFOAM with SIMPLE algorithm) and finite difference method (coarse and finer grid). The mutual agreement between the finer grid for FDM and in-house FVM code depends on the angle α, with best results (smallest solution differences) achieved for angles close to



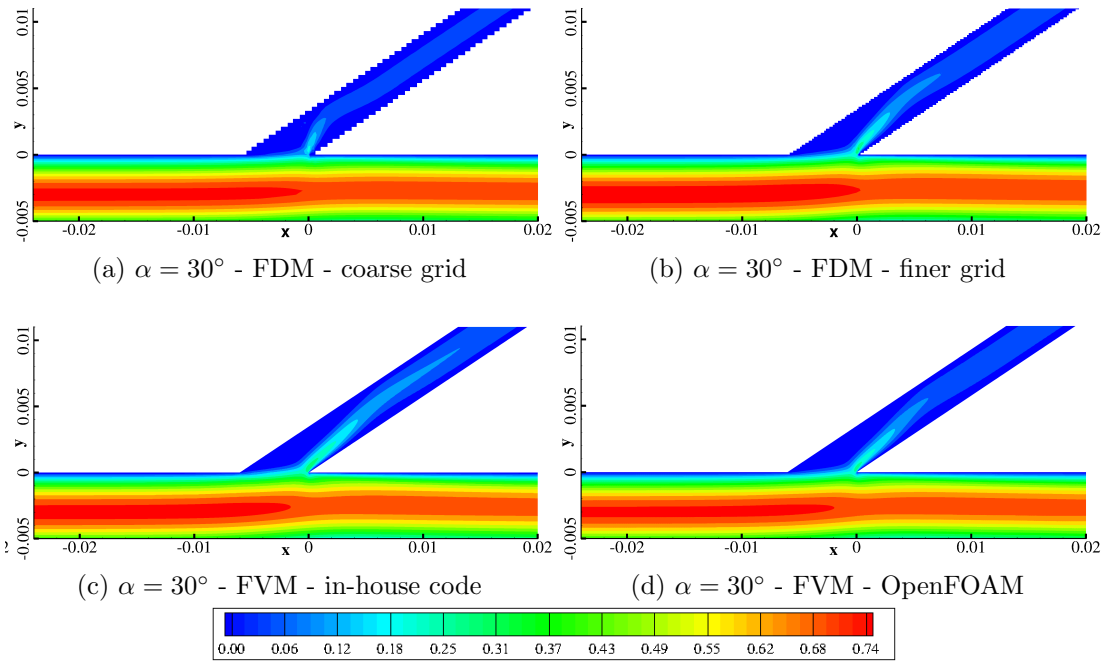
(a) $\alpha = 30°$ - FDM - coarse grid

(b) $\alpha = 30°$ - FDM - finer grid

(c) $\alpha = 30°$ - FVM - in-house code

(d) $\alpha = 30°$ - FVM - OpenFOAM

Figure 4: Pressure field in detail for the case $\alpha = 30°$, different solvers and grids.



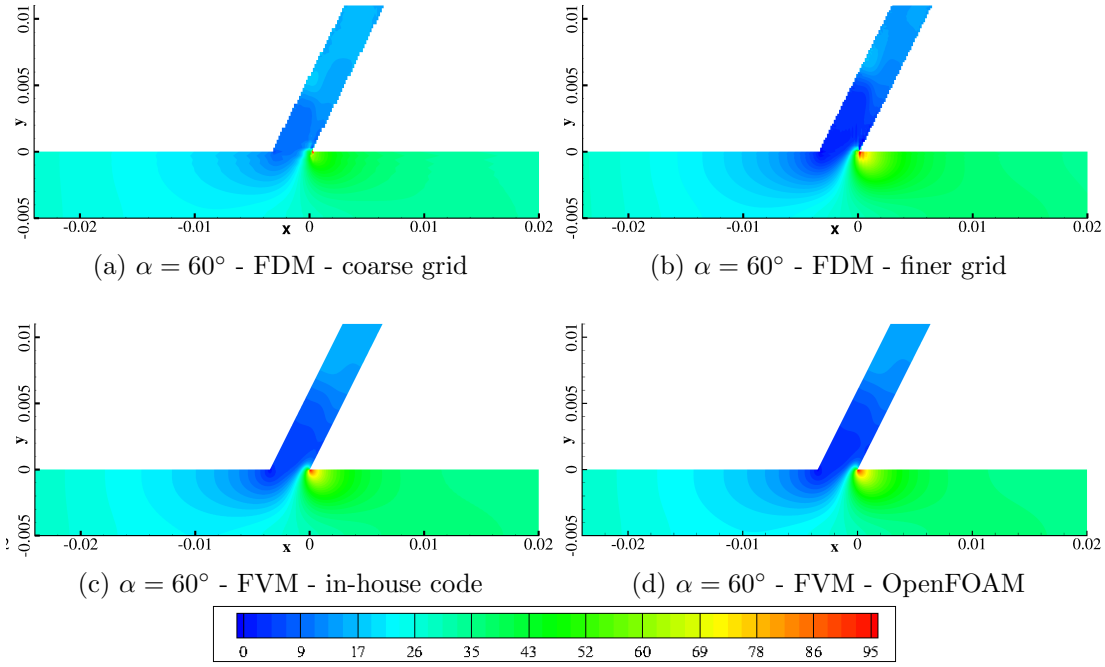
(a) $\alpha = 30°$ - FDM - coarse grid

(b) $\alpha = 30°$ - FDM - finer grid

(c) $\alpha = 30°$ - FVM - in-house code

(d) $\alpha = 30°$ - FVM - OpenFOAM

Figure 5: Velocity magnitude in detail for the case $\alpha = 30°$, different solvers and grids.



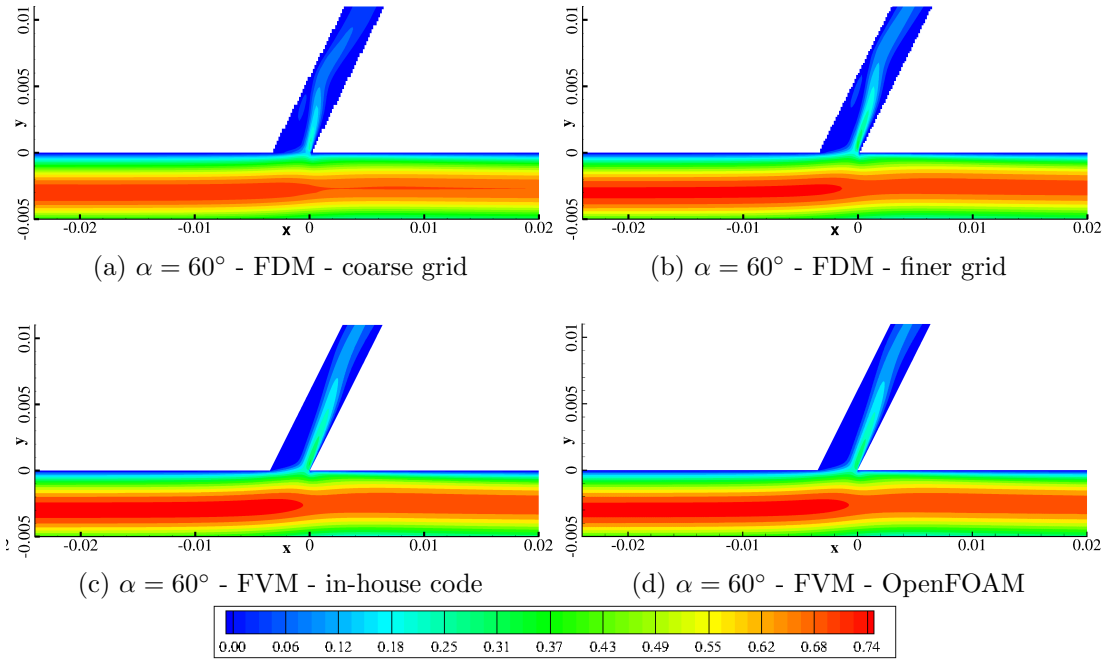
(a) $\alpha = 60°$ - FDM - coarse grid

(b) $\alpha = 60°$ - FDM - finer grid

(c) $\alpha = 60°$ - FVM - in-house code

(d) $\alpha = 60°$ - FVM - OpenFOAM

Figure 6: Pressure field in detail for the case $\alpha = 60°$, different solvers and grids.



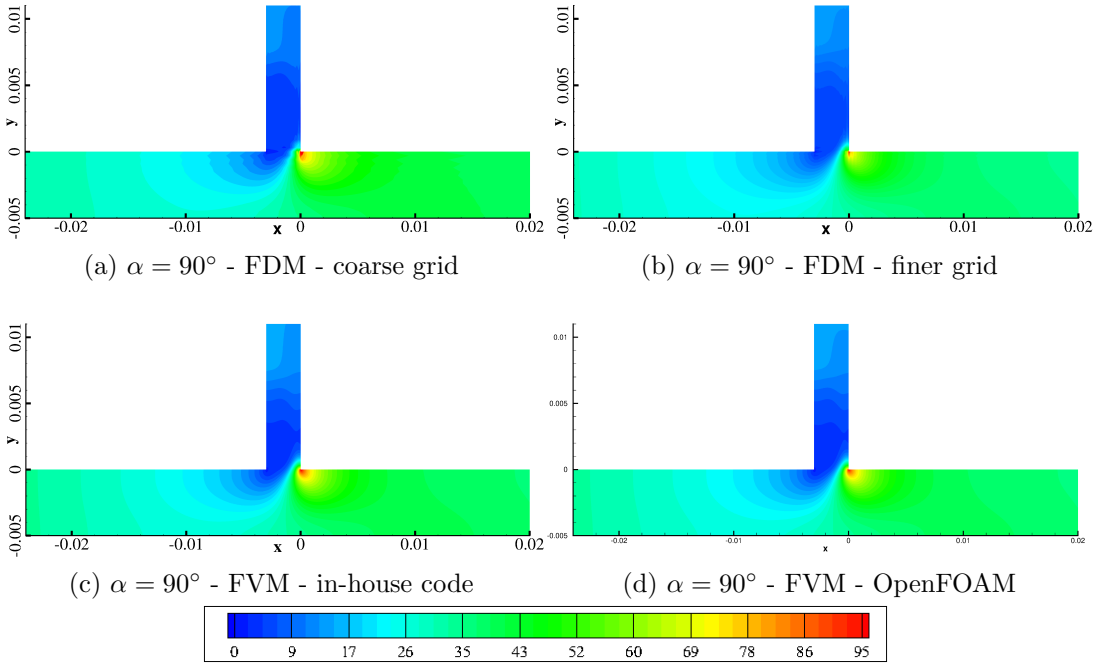
(a) $\alpha = 60°$ - FDM - coarse grid

(b) $\alpha = 60°$ - FDM - finer grid

(c) $\alpha = 60°$ - FVM - in-house code

(d) $\alpha = 60°$ - FVM - OpenFOAM

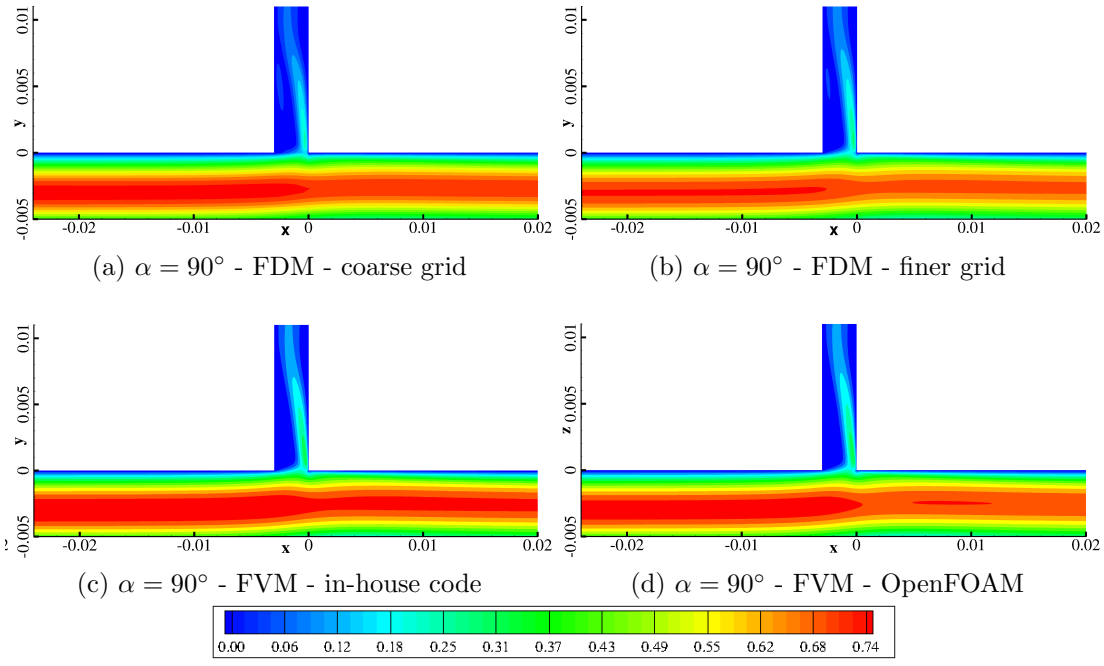
Figure 7: Velocity magnitude in detail for the case $\alpha = 60°$, different solvers and grids.



(a) $\alpha = 90°$ - FDM - coarse grid (b) $\alpha = 90°$ - FDM - finer grid

(c) $\alpha = 90°$ - FVM - in-house code (d) $\alpha = 90°$ - FVM - OpenFOAM

Figure 8: Pressure field in detail for the case $\alpha = 90°$, different solvers and grids.



(a) $\alpha = 90°$ - FDM - coarse grid (b) $\alpha = 90°$ - FDM - finer grid

(c) $\alpha = 90°$ - FVM - in-house code (d) $\alpha = 90°$ - FVM - OpenFOAM

Figure 9: Velocity magnitude in detail for the case $\alpha = 90°$, different solvers and grids.

$\alpha = 90°$, while in the case $\alpha = 30°$ (and $\alpha = 150°$, not shown here) the differences are more pronounced.

5. Conclusions

A new numerical code was developed based on a finite difference method using the immersed boundary approach, which was applied to the numerical simulation of the flow of viscous incompressible fluid in planar branching channels.

The numerical results were presented in this work showed that the results obtained by the newly developed code are comparable to the results provided by the previously used code based on the finite volume method and also to the results from the open-source package OpenFOAM.

The dependence of the immersed boundary method on the grid resolution was found, especially during numerical simulations in channels with oblique branching. In the case of a perpendicular connection, the differences between coarser and finer grids were not so large. Although the results obtained on the coarse and finer grids are qualitatively very similar (showing the same flow structure), some quantitative parameters (such as the maximum velocity or discharge) may differ.

In the presented comparison, a simple pressure-based setup was chosen, where the flow is controlled only by the prescribed pressure differences between the inlet/outlet boundaries of the channel branches. Such setup is very sensitive to the numerical method, the grid structure, and the way the boundary conditions are imposed. This sensitivity is due to the fact that the flow in the channel branches is unknown in advance, and the flow field develops only due to the pressure difference. In this context, the agreement between the numerical predictions of the three considered methods and codes can be evaluated as satisfactory.

Our future work will focus on the extension of the presented comparison for unsteady flows and non-Newtonian fluids, which is crucial for the intended investigation of various biomedical applications.

Acknowledgements

This work was supported by the grant SGS22/148/OHK2/3T/12 and partly by the *Praemium Academiae* of Š. Nečasová.

References

[1] Beneš L., Louda P., Keslerová R., Kozel K. Štigler J.: Numerical simulations of flow through channels with T-junction, Journal of Applied Mathematics and Computation **219** (2013) 7225–7235.

[2] Bodnár T., Sequeira A.: Numerical simulation of the coagulation dynamics of blood, Computational and Mathematical Methods in Medicine **9** (2008) 83–104.

[3] Bodnár, T., Keslerová, R., Lancmanová, A.: Numerical Investigation of Incompressible Fluid Flow in Planar Branching Channels *Recent Advances in Mechanics and Fluid-Structure Interaction with Applications*. Basel: Birkhäuser (2022) 95–126.

[4] Chorin, A. J.: A numerical method for solving incompressible viscous flow problem. Journal of Computational Physics **135** (1967) 118–125.

[5] Keslerová, R., Trdlička, D.: Numerical solution of viscous and viscoelastic fluids flow through the branching channel by finite volume scheme, Journal of Physics: Conference Series **633** (2015).

[6] Keslerová R., Trdlička D., Řezníček H.: Numerical simulation of steady and unsteady flow for generalized Newtonian fluids, Journal of Physics, Conference Series **738** (2016).

[7] Keslerová R., Řezníček H., Padělek T.: Numerical modelling of generalized Newtonian fluids in bypass tube, Advances in Computational Mathematics **45** (2019) 2047–2063.

[8] Lancmanová A., Bodnár T., Keslerová R.: Numerical Validation of a Simple Immersed Boundary Solver for Branching Channels Simulations, In: D. Šimurda and T. Bodnár (Eds.) *Proceedings Topical Problems of Fluid Mechanics 2022*, IT CAS, Prague 2022, 127–134.

[9] MacCormack R. W.: The effect of viscosity in hypervelocity impact cratering, AIAA (1969) 69–354.

[10] Mittal R., Iaccarino G.: Immersed boundary method, Annual Review of Fluid Mechanics **37** (2005) 239–261.

[11] Moukalled F., Mangani L., Darwish M.: *The Finite Volume Method in Computational Fluid Dynamics*, Springer, Heidelberg, 2016.

[12] Patankar S. V., Spalding D. B.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer **15** (1972) 1787–1806.

[13] Peskin C. S.: The immersed boundary method, Acta Numerica (2002) 479–517.

[14] Winter O., Bodnár T.: Simulations of viscoelastic fluids flows using a modified log-conformation reformulation, In: D. Šimurda and T. Bodnár (Eds.) *Proceedings Topical Problems of Fluid Mechanics 2017*, IT CAS, Prague 2017, 321–328.