STEADY INCOMPRESSIBLE FLOW THROUGH A BRANCHED CHANNEL

Anna Lancmanová 1, Tomáš Bodnár 1
1 Faculty of Mechanical Engineering, Czech Technical University in Prague, Karlovo Náměstí 13, 121 35 Prague 2, Czech Republic

Abstract
This contribution presents some of the first results of a newly developed simple numerical solver aimed to study the flow properties and behavior in branching channel. The 2D incompressible fluid flow is simulated using Navier Stokes equations, solved by a finite-difference scheme on a Cartesian grid. The channel geometry is represented by a simple immersed boundary method implementation. A series of essential tests was performed for the selected geometry to evaluate the overall behaviour of the model and its response to different boundary conditions, geometry and grid settings.

Keywords: branching channel, incompressible Navier-Stokes, finite difference, pressure correction

1 Introduction
This work is motivated by our long term aim to study the airflow in the human upper airways. It is a highly complex problem and its mathematical modelling and numerical simulation constitutes of many difficult and challenging tasks. This is why as a first step in this long term project it was decided to build a simple demonstrator toolbox that will enable us to study the efficiency, suitability and influence of different geometries, models, computational setups and numerical solvers on the model outcomes.

In majority of cases the airflow in human airways can be considered as incompressible, therefore the mathematical model is based on incompressible Navier-Stokes equations. Standard approach using a finite volume method for similar case can for example be found in [4]. In order to keep the problem solution simple, the immersed boundary method was chosen here to represent the (possibly) complex geometry of branching airways. The discretisation on Cartesian grid allows to easily and efficiently implement various finite-difference methods to numerically solve the governing system. Also the implementation of boundary conditions is quite straightforward under such setup. For this initial study only the steady, laminar flow was considered.

The aim of this work is to verify the overall functionality of the new code and to show how different model and solver parameters will affect the solution.

2 Mathematical Model
Governing equations
The incompressible Navier-Stokes equations can be written in a conservative vector form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u},$$

where $\rho$ is the (constant) density, $\nu$ is the (constant) kinematic viscosity and $p$, $\mathbf{u}$ is the pressure and velocity field respectively. The velocity field obeys the incompressibility (divergence-free) constraint ($\nabla \cdot \mathbf{u} = 0$).

Computational domain
The two-dimensional (2D) computational domain represents a rectangle in $x-z$ plane with dimensions $4 \times 2$. The numerical simulations were performed on a structured, orthogonal (Cartesian) grid with different number of cells. The standard grid had $640 \times 320$ cells. Other coarser and finer grids were used to assess the effect and efficiency of the discretization refinement.

The coordinate system has been chosen to have the origin in the middle of the left side ($x-z$ coordinates). The bounded computational domain is shown on Fig.1.
Initial and boundary conditions

In this initial series of simulations only steady cases are considered. Different computational cases defined by setting different boundary conditions, namely the pressures at the channel inlets and outlets.

- **Inlet** (at $x = 0$) – The constant pressure $p_{in} = 0$ is prescribed. Homogeneous Neumann condition is used for all velocity components.

- **Outlet** $a$ and $b$ (at $x = l_x$) – The constant pressure $p_a$ and $p_b$ is prescribed at the corresponding parts of the boundary. Homogeneous Neumann condition is used for all velocity components.

- **Walls** – On the channel walls no-slip conditions are used for the fluid velocity, i.e. the velocity vector is set to $u = (0, 0)$.

Due to immersed boundary implementation, the velocity is set zero inside the part of the domain occupied by the solid material, so no special treatment is needed on the channel walls for pressure or velocity.

3 Numerical Methods

The governing system of incompressible Navier-Stokes equations is discretized using finite-difference method and solved by a modified Lax-Friedrichs scheme with reduced (and adjustable) numerical dissipation. The pressure-velocity coupling is carried out by a pressure correction method described below.

The pressure correction method is an iterative procedure in which the components of intermediate velocity field (disobeying the divergence-free constraint) is first calculated in each iteration, subsequently corrected using the pressure correction to satisfy the continuity equation. The calculation procedure is shown in the algorithm diagram in Fig. 2 (for details see e.g. [2]).

The rudimentary approach of pressure correction methods is decoupling computation of the pressure field from the velocity field. In basic form of this method, the intermediate velocity field $u^*$ is solved first

$$\frac{u^* - u^n}{\Delta t} = -\nabla \cdot (u \otimes u)^n - \frac{1}{\rho} \nabla p^n + \nu \Delta u^n \quad \Rightarrow \quad u^*. \quad (2)$$

In this step the Lax-Friedrichs method (which contains an explicit Euler scheme of time discretisation - see eq. (3)) is used in a modified form, which allows to reduce the embedded numerical viscosity of the scheme (using the adjustable parameter $\zeta$). This scheme is very simple, well known (although definitely not the best one) and was chosen as a reference for future comparison with more advanced schemes. The influence of the parameter $\zeta$ and a more detailed description of the pressure correction method used can be found in [5], [6].
\[ u_{i,j}^* = (1 - \zeta_u) u_{i,j}^n + \zeta_u \left( \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n}{4} \right) + \Delta t \left[ -\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h} u_{i,j}^n + \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h} \right] - \frac{1}{\rho} \left( \frac{p_{i+1,j}^n - p_{i-1,j}^n}{2h} + \nu \left( \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h^2} \right) + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h^2} \right) \]

(3)

The intermediate velocity estimate \( u^* \) does not satisfy the continuity equation (incompressibility constraint). Therefore the final values of velocity \( u^{n+1} \) and pressure \( p^{n+1} \) must be corrected by adding corrections \( u', p' \) to the intermediate values

\[ u^{n+1} = u^* + u', \quad p^{n+1} = p^n + p', \]

where velocity and pressure corrections are given as

\[ u' = -\frac{\Delta t}{\rho} \nabla p', \quad \Delta p' = \frac{\rho}{\Delta t} \nabla \cdot u^*. \]

(5)

The schematic picture of the pressure correction algorithm is shown in the Fig. 2.

![Algorithm diagram for pressure correction method.](image)

Figure 2: Algorithm diagram for pressure correction method.

In the pressure correction algorithm a Poisson equation should be solved to obtain the pressure corrections field \( p' \). The discretization of the Laplacian is made on a compact nine-point stencil, with weights shown in the schematic picture in Fig. 3. In our case, the general quantity \( \phi \) represents the pressure correction \( p' \) and \( f \) stands for right side of the equation for pressure correction, see Eq. 5. The whole discrete Poisson equation for the pressure correction leads to a system of linear algebraic equations. It is solved by the linear SOR solver with relaxation parameter \( \omega = 1.75 \). More details can be found in [3], [7] or [1].

![Compact nine-point stencil for the Poisson equation and Laplace operator.](image)

Figure 3: Compact nine-point stencil for the Poisson equation and Laplace operator.

4 Numerical Simulations and Results

The hereafter presented simulations are performed to verify the qualitative behaviour of the simulation outputs and their sensitivity to various parameters. To present the results, the simulations were divided into two larger groups. In the first group are simulations with fixed sharp-edged channel splitting, showing different flow regimes depending on the various pressures set on the boundaries. The second group of simulations deals with changes of the shape of the channel branching, where the sharp edge is replaced by a circular geometry with different radii.
Branched pipe with a sharp edge

The first solved problem is flow through a branched pipe, where the two branches are separated by a sharp wadge. Different flow regimes are achieved by setting different pressure values (and thus pressure drops) at different parts of the boundary. For all simulations the inlet pressure $p_{in} = 0$ (at $x = 0$). Then different pressures $p_a$, $p_b$ were set in branches $a$ and $b$ (at position $x = l_x$). Complete overview of test cases is summarized in Tab. 1 and Fig. 6. Three basic cases were considered:

(a) symmetric outlet pressure $p_a = c$, $p_b = c$,

(b) anti-symmetric outlet pressure $p_a = c$, $p_b = -c$,

(c) zero outlet pressure $p_a = 0$, $p_b < 0$.

Typical results obtained for such pressure configurations are shown in the Fig. 4.

These cases were selected to demonstrate the basic effect of the pressure drop on the flow rate and the outlet pressure ratio between the branches on the flow splitting. The effect of the outlet pressure magnitude (and thus of pressure drop with respect to inlet) is demonstrated in the Fig. 5. As expected the increase of the flow rate is observed for larger pressure drops, but the dependence itself is non-linear.

Figure 4: Total velocity in dependence on pressure $p_a, p_b$.

Figure 5: Volumetric flow rate in dependence on pressure $p_a, p_b$. 

![Diagram](https://example.com/diagram1.png)  
![Diagram](https://example.com/diagram2.png)  
![Diagram](https://example.com/diagram3.png)
The total inflow and outflow was checked along the whole boundary of the computational domain to verify the mass conservation of the numerical model. This is especially important for the immersed boundary method implemented in the code, in which the velocities in cells outside the channel (outside the fluid) are set to zero.

A very small difference (of order at maximum $10^{-7}$) between inflow and outflow is visible in Tab. 1. Such accuracy is satisfactory (given the extreme simplicity of the methods used) and the differences can be attributed to numerical inaccuracies and the use of an immersed boundary method.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Volumetric flow rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>case</td>
<td>$p_a$</td>
</tr>
<tr>
<td>I</td>
<td>20</td>
</tr>
<tr>
<td>II</td>
<td>10</td>
</tr>
<tr>
<td>III</td>
<td>5</td>
</tr>
<tr>
<td>IV</td>
<td>-20</td>
</tr>
<tr>
<td>V</td>
<td>-10</td>
</tr>
<tr>
<td>VI</td>
<td>-5</td>
</tr>
<tr>
<td>VII</td>
<td>0</td>
</tr>
<tr>
<td>VIII</td>
<td>0</td>
</tr>
<tr>
<td>IX</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Volume flow rate in dependence on pressure settings.

Branched pipe with rounded edge

The second series of tests deals with modified geometry of the channel, where the sharp wedge separating the branches was replaced by a rounded one, with different radii. The goal was to test the effect of the rounding radii to attained flow rate for the same pressure drops. A secondary subset of these simulations was performed on grids of various density (different cell sizes) to assess the grid independence of results and also the computational efficiency of the whole code.

For these tests three radii were chosen to observe the effect of wedge rounding. The center of rounding circle is calculated based on the selected radius to satisfy the tangential connection to the wall edge (surface boundary smoothness). Schematic picture of the channel with rounded branching is shown in Fig. 9 (lower left corner), where red circle marks the rounded edge.

As expected, the general structure of the flow in a channel with rounded branching remains the same as for the case with sharp wedge. This is clear from comparison of the velocity fields in Fig. 7 (rounded branching) and Fig. 4 (sharp branching).
The effect of the rounding radius is however substantial, which is visible in the Fig. 8 showing again the velocity field (in the same color scale) for four different radii $r = 0.05, 0.1, 0.15, 0.2$. Evidently higher velocities develop in the channel (for the same pressure drop) when the radius is larger.

The increase of the resulting flow rate, in dependence on the increased radius of branching, can be assessed from the column graphs shown in Fig. 9. This tendency can probably be attributed to increase of the low velocity region close to stagnation point on the radius which reduces the viscous losses in the flow. This effect is also enhanced by shorter branches which also reduces the friction losses.
In order to assess the dependency of the results on the grid spacing, several coarser and finer grids were tested. The finer grid always had twice as many grid points in each direction as the coarser one (except the last one). The resulting volume flow rates obtained on different grids for the rounded edge case with $r = 0.1$ are shown in the Tab. 1.

<table>
<thead>
<tr>
<th>Grid density</th>
<th>Volumetric flow rate for $r = 0.1$</th>
<th>Input</th>
<th>Output</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>a</td>
<td>b</td>
<td>outflow</td>
</tr>
<tr>
<td>161x81</td>
<td>0.54191</td>
<td>0.27119</td>
<td>0.27112</td>
<td>0.54231</td>
</tr>
<tr>
<td>321x161</td>
<td>0.50223</td>
<td>0.25120</td>
<td>0.25103</td>
<td>0.50223</td>
</tr>
<tr>
<td>641x321</td>
<td>0.49752</td>
<td>0.24873</td>
<td>0.24879</td>
<td>0.49752</td>
</tr>
<tr>
<td>961x481</td>
<td>0.49347</td>
<td>0.24671</td>
<td>0.24676</td>
<td>0.49347</td>
</tr>
</tbody>
</table>

Table 2: Volume flow rate in dependence on grid density for $r = 0.1$ and $p_a = -10, p_b = -10$.

The dependence of the flow rate on the grid density is plotted in the Fig. 10 (a). Based on this and also on the time needed for solution on grids of different size (see Fig. 10 (b)), it was decided to use the grid with $641 \times 321$ points as a standard for all simulations. On this grid the rounding radius can be represented with sufficient accuracy, as it is demonstrated in the 10 (c).
5 Conclusions & Remarks

The numerical simulations presented in this paper have demonstrated the expected behaviour and properties of the newly developed numerical code. From the results obtained during the testing, the following conclusions can be drawn.

- The chosen Lax-Friedrichs scheme is robust accurate enough for the steady flow simulations at low velocities (low Reynolds numbers). For unsteady flow simulations and faster flows it will be necessary to implement some more accurate (in space and time) discretization.
- The immersed boundary method seems to perform quite well despite of the very simple implementation chosen in this code. The question is whether it will be good enough also for unsteady simulations.
- The computational setup represented by imposing just pressure on the external boundaries of the channel has worked well. The only problem seems to be with the homogeneous Neumann conditions imposed on velocity, which might be too restricting, leading to flow deviations close to outlet boundaries. In this view also the prescription of the constant pressure across the outer boundaries of both branches might be relaxed.

The future development of this code will focus on its testing for unsteady flows and implementation of other numerical schemes.

Acknowledgment

The financial support for the present work was partly provided by the Czech Technical University, Faculty of mechanical engineering under the Grant No. SGS19/154/OHK2/3T/12 and partly by the European Regional Development Fund-Project “Center for Advanced Applied Science” No.CZ.02.1.01/0.0/0.0/16019/0000778.

References


