

Numerical Methods in Modelling of Red Blood Cellflow Behaviour Through a Specific Stacking Pattern

*OROVWUJE, Paul Stephen¹, OSAIDE, Stella Eguono²

¹Mathematics Department - Edwin Clark University, Kiagbodo, Delta State

²Department of Biology - College of Education, Warri, Delta State

*Corresponding Author

DOI: <https://doi.org/10.51583/IJLTEMAS.2024.130918>

Received: 19 September 2024; Revised: 07 October 2024; Accepted: 10 October 2024; Published: 21 October 2024

Abstract: The dynamics of red blood cells (RBCs) is one of the major aspects of the cardiovascular system that has been studied intensively in the past few decades. Using computational fluid dynamics, complex nonlinear fluid flows have been modeled. The dynamics of biconcave RBCs are thought to have major influences in cardiovascular diseases and other problems associated with cardiovascular flow behaviour, and the determination of blood rheology and properties. Most reported computational models have been confined to the behaviour of a single RBC in 2-dimensional domains, under physiological flow conditions. This work investigates a particular stacking pattern in analyzing the RBC flow behavior under physiological flow conditions, using the D2Q9 lattice Boltzmann numerical method created using Matlab. Prior to the analysis the Matlab script was benchmarked using the Poiseuille flow and the flow around the cross-section of a cylinder, after which the accuracy of the method used was determined. The benchmarks showed that the lattice Boltzmann code had minimal error. The accuracy was determined using the data obtained from Matlab and a created excel program. It also showed that the lattice Boltzmann method was of the first order, which corresponds with results existing in literatures. The analysis of the stacking pattern showed how RBC flows through the chosen stacking pattern, and the results are shown.

Keyword: lattice Boltzmann method, Red Blood Cell, Fluid Flow.

I. Introduction

Blood is a specialized connective tissue composed of cells suspended in a fluid matrix called plasma. Its primary function is to transport vital substances, including oxygen, carbon dioxide, nutrients, waste products, and hormones, throughout the body. Blood cells encompass erythrocytes (red blood cells), leukocytes (white blood cells), and thrombocytes (platelets). Erythrocytes exhibit a distinctive discoid shape with a thickened rim and a thinner central region.

[14] Introduced, a strain energy function to characterize the elastic properties of the human red blood cell membrane. This function incorporates multiple elastic moduli to account for the membrane's complex behavior under deformation. The model successfully predicts the sphering of red blood cells in hypotonic solutions and provides insights into phenomena such as sieving and micropipette experiments.

Human adults typically possess a blood volume of 4-6 liters, maintained in circulation by the heart's pumping action [10]. The heart consists of four chambers: two atria and two ventricles. Valves ensure unidirectional blood flow between chambers. The circulatory system comprises two interconnected circuits: the pulmonary circuit, which exchanges gases between the lungs and blood, and the systemic circuit, which delivers oxygen and nutrients to tissues while removing waste products [10].

The Navier-Stokes equations serve as the fundamental framework for modeling blood flow. While complex, these equations can be simplified through appropriate assumptions, leading to linearized models. Early computational fluid dynamics (CFD) studies employed these simplified models to investigate blood flow patterns. Pioneering work by Harlow's group at Los Alamos National Laboratory in the 1950s and 1960s advanced CFD techniques for two-dimensional fluid flow [1]. The Panel method, introduced by [6, 7], represented a significant step forward in three-dimensional CFD simulations.

II. Materials and Methods

The lattice Boltzmann method (LBM) evolved from lattice gas automata, initially introduced in 1976 by [4] and later expanded upon by [3]. LBM's versatility in addressing complex fluid dynamics problems, including red blood cell (RBC) flow, is attributed to its adaptability to various dimensions and accuracies. Its applications encompass 1 -3 dimensional simulations with varying levels of precision. This study aims to determine numerical solutions for Poiseuille flow (driven by pressure gradient or fixed inlet velocity) and flow around a cylindrical cross-section (using Stokes' stream function) through LBM. [11, 12, 13]

A fundamental step involves reviewing the lattice Boltzmann model for the incompressible Navier-Stokes equation. A key aspect is the explicit elimination of terms proportional to the square of the Mach number (M^2) to mitigate density fluctuations inherent in traditional LBM. The Chapman-Enskog procedure is employed to derive the incompressible Navier-Stokes equation from the incompressible LBM. These methodologies, as detailed by various researchers, serve as essential tools for developing a MATLAB implementation of the lattice Boltzmann equation.

III. The Lattice Boltzmann Method

The Boltzmann equation, underpinning the LBM, relies on the Stosszahl Ansatz, or molecular chaos assumption, to significantly reduce computational cost [12]. This assumption, while sacrificing detailed particle-level information, proves adequate for fluid dynamics simulations operating at scales far exceeding the molecular level.

The usage of Boltzmann's so called Stosszahl Ansatz, which is an assumption of molecular chaos, will be a much computationally cheaper solution, [13]. The usage of the Stosszahl Ansatz results directly in the following equation

$$N_i(\vec{r} + \vec{v}_i \Delta t, t + \Delta t) = N_i(\vec{r}, t) + \Omega_i(N_i) \quad 3.0$$

This results in a loss of detail of the dynamics of each separate particle, which saves computational time. For some cases the loss of detail can be a disadvantage. However, for fluid dynamics applications, this assumption is a valid one since the scale of the flow is much bigger than the molecular scale [12].

The lattice Boltzmann method is a method based on the lattice gas automata. The LBM is capable of solving complex three-dimensional problems on fluid dynamics. The main equation when using the lattice Boltzmann method is given by

$$N_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = N_i(\vec{x}, t) + \Omega_i(N_i(\vec{x}, t)) \quad 3.1$$

in which the collision operator $\Omega_i(N_i(\vec{x}, t))$ has been defined as

$$\Omega_i(N_i(\vec{x}, t)) = -\frac{1}{\tau} [N_i(\vec{x}, t) - N_{i,eq}(\vec{x}, t)] \quad 3.2$$

The collision factor $\Omega_i(N_i(\vec{x}, t))$ is required to satisfy the law of conservation of mass and momentum at each lattice node. Hence, it holds that the following rule is applicable for Ω_i , ie.

$$\sum_i \Omega_i = 0 \quad 3.3$$

Considering equation (3.62), τ stands for the relaxation time which is a measure for how fast the collisional operator relaxes from the actual to the equilibrium configuration. The viscosity is related to the relaxation time as well as the particle velocity in a certain direction according to

$$\nu = (\tau - 0.5)c_i^2 \quad 3.4$$

ν is the kinematic viscosity, τ is the relaxation time and c_i is a set of variables describing the particle velocity. The pressure and density are also related to each other. This relationship is represented by

$$p = \rho c_i^2 \quad 3.5$$

Inserting equation (3.2) into equation (3.1) gives the complete equation for the lattice Boltzmann method.

$$N_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = N_i(\vec{x}, t) - \frac{1}{\tau} [N_i(\vec{x}, t) - N_{i,eq}(\vec{x}, t)] \quad 3.6$$

$c_i (i = 1, 2, \dots, M)$ is as earlier define with M as the number of direction of the particle velocities at each node.

$N_i (i = 1, 2, \dots, M)$ is a set of variables describing the particle occupation, again with M being the number of directions. The number of variables depends on the type of model one uses.

The DnQm Models

The problems that one can encounter are of different dimensions and complexities. There are several lattice Boltzmann models which can be used that are suitable for each dimension. These models are all of the form DnQm, in which n stands for the dimension and m stands for the number of speeds of the model. These speeds are the numbers of discrete velocity vectors used to create the grid. Different models are given for two and three dimensions [9].

IV. Results and Discussion

The lattice Boltzmann method is used to simulate the RBCs flow through a specific stacking pattern of boundaries. This is done by creating a program to execute the lattice Boltzmann equation in Matlab.

4.1 The Computational Flow Chart

The computational program consists of four major parts which will be explained further. The first step starts with the creation of the grid which is done by defining the dimensions of the grid, i.e. l_x and l_y , being the lengths of the square grid (i.e. $l_x = l_y$). Next, some essential parameters are set a priori of the main program. These are the relaxation time, viscosity, the density and the weight factors. Initializing the LB-matrix is the last part of this step and is done by creating the LB- matrix using the internal Matlab function `repmat` (the LB-matrix is a matrix having the entire lattice Boltzmann functions in it). `Repmat(A,[M N P ...])` tiles the array A to produce a multi-dimensional array composed of copies of A . After the LB-array has been created it has to be initialized which can be done in several ways, the one used in this work is by stating that the LB-array is equal to the equilibrium terms

The second part of the code is to generate the boundaries. This is done by creating a Boolean variable in which 1 means that it is a boundary and 0 means that it is not a boundary. An advantage of this is that it enables the opportunity to create any boundary one desires.

The third step is to execute the LBM in a *while loop* to create a velocity profile. The first part of the *while loop* includes the creation of the nine LB-functions which are initialized in step 1. After the macroscopic parameters are calculated, being the x-velocity, y-velocity and the density. Those are needed to determine the nine equilibrium terms. The last part of the *while loop* is to enlarge the time by one unit so the *while loop* runs again until the maximum steps are used.

The fourth and final step is to create a vector field which represents the velocity field. This is done by using the internal Matlab function *quiver*. *Quiver* (X, Y, U, V) plots velocity vectors as arrows with components (u,v) at the points (x,y), [2].

4.2 Benchmarks

This is done by benchmarking the program and letting the program execute problems which can be evaluated analytically. In this case two benchmarks have been done. The first is the **Poiseuille flow** and the second is the **flow around the cross-section of a cylinder** using Stokes' stream function.

4.2.1 Poiseuille Flow

The first benchmark that has been executed is the Poiseuille flow. This is a laminar flow through two solid boundaries. The laminar flow is the normal condition for blood flow throughout most of the circulatory system and it is characterized by concentric layers of blood moving in parallel down the length of a blood vessel. Generally the body RBC flow is laminar. However, under condition of high flow, particularly in ascending aorta with high Reynolds number laminar flow can be disrupted and become turbulent. When this occurs, the RBC does not flow linearly and smoothly in adjacent layers, but instead the flow can be described as being chaotic.

4.2.1.1 The Analytical Solution

To obtain the velocity profile of a Poiseuille flow the general Navier-Stokes equations for an incompressible medium is considered. Recall equation (3.1)

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho \vec{u} \cdot \nabla \vec{u} = -\nabla \vec{p} + \mu \nabla^2 \vec{u} + \vec{F}_{ext} \quad 4.1$$

First of all when dealing with incompressible flow, the Navier-Stokes equation can be simplified into

$$\nabla \vec{p} = \mu \nabla^2 \vec{u} + \vec{F}_{ext} \quad 4.2$$

In this particular case there is no external force, meaning that the $\vec{F}_{ext}$ term is zero. Also due to the fact that it is a symmetric problem, equation (4.2) can be simplified even further into

$$\frac{\partial \vec{p}}{\partial x} = \mu \frac{\partial^2 \vec{u}}{\partial y^2} \quad 4.3$$

Solving this equation for $\vec{u}(i)$ is an easy differential equation, the result is, with $\frac{\partial \vec{p}}{\partial x}$ taken as constant

$$\vec{u}(i) = \frac{1}{2\mu} \frac{\partial \vec{p}}{\partial x} y^2 + C_1 y + C_2 \quad 4.4$$

Boundary conditions are needed to obtain for C_1 and C_2

Boundary condition 1: $\vec{u}(0) = 0$ gives $C_2 = 0$ (non-slip condition)

Boundary condition 2: $\vec{u}(D) = 0$ gives $C_1 = -\frac{1}{2\mu} \frac{\partial \vec{u}}{\partial x} D$

Inserting the previously found C_1 and C_2 in the equation for $\vec{u}$ and simplifying the result gives

$$\vec{u} = -\frac{1}{2\mu} \frac{\partial \vec{u}}{\partial x} (Dy - y^2) \quad 4.5$$

In this equation y is the distance from the bottom of the plate to a point between the plates, and D is the distance between the bottom and the top of the plate, [1].

From equation (4.5) one can see that the maximum velocity occurs precisely in the middle of the two plates at $y = \frac{D}{2}$. This is done by solving $\frac{\partial \vec{u}}{\partial y} = 0$. At the solid boundaries the velocity is zero.

This is in line with the no-slip boundary condition.

Taking $i = 1, 2, 3, \dots, 101$, $D = 1$, $\frac{\partial \vec{u}}{\partial x} = 1$, $\mu = 1$, $y = 0, 0.01, 0.02, \dots, 1$ the analytically generated values are $\hat{u}(i) = 0, -0.00495, -0.0098, -0.01454999999, -0.0192000000000, -0.02375 \dots$

4.2.1.2 Results of the Analytical Poiseuille Flow

The results of the analytically determined Poiseuille flow are represented in figure 1 which has been made creating a Matlab code. The velocity in the upward direction is defined as positive making the velocity in figure 1 positive as well. In practice this means that the pressure gradient is positive.

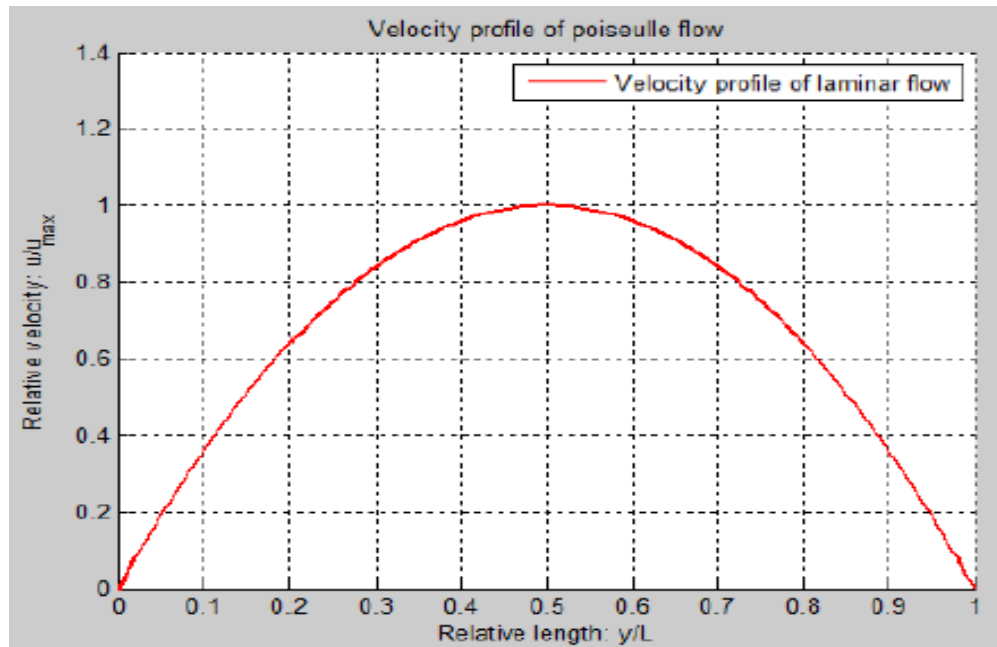


Figure 1: Visualization of the analytical solution of the Poiseuille flow. In the Matlab program the inflow and outflow boundaries have been made periodic. This has been done by stating that, what flows in equals what flows out, making the boundaries periodic.

4.2.1.3 The Computationally Determined Solution

For the Matlab script to give the Poiseuille flow, the boundaries have to be set properly, as shown above. Since the D2Q9 model is used, the boundaries are set at both ends of the horizontal axis. The script export the obtained data to an excel sheet where the values are saved. The data from excel are imported in Matlab where further calculations are executed.

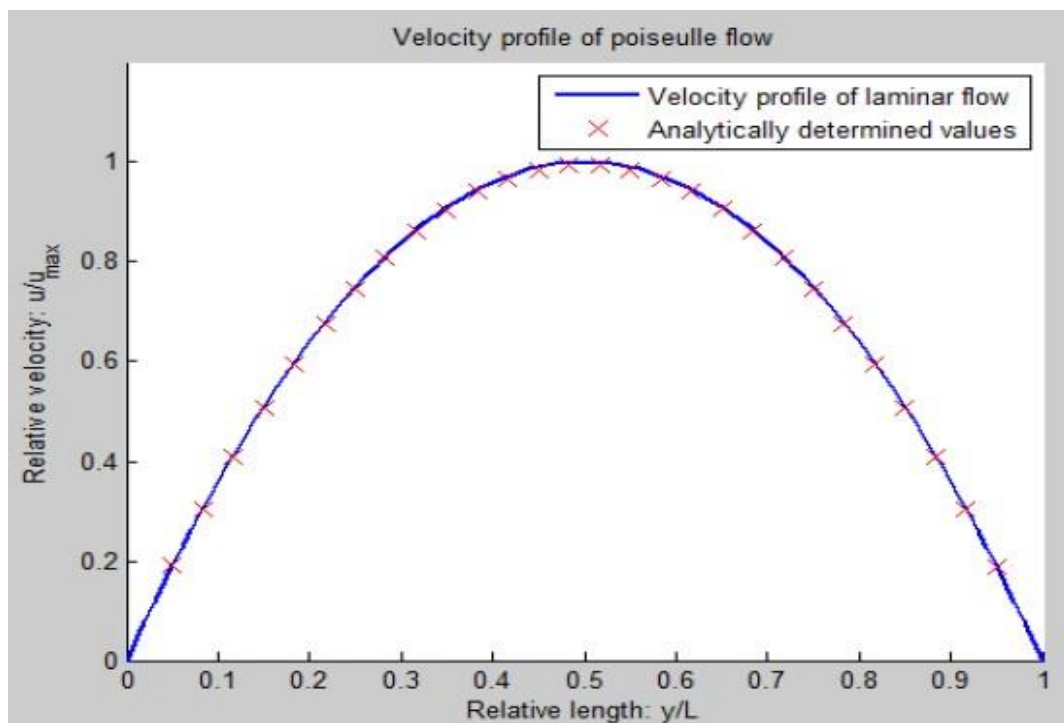


Figure 2: The analytical solution together with the determined values of the Poiseuille flow using the half-way bounce back method

4.2.1.4 Comparing the Two Results

In a secluded Matlab program the exported values of the Poiseuille flow, determined with the lattice Boltzmann method, are compared with the analytical solution. The analytical solution is plotted as a graph and the determined values are plotted in the plot as x-marks. As visible in figure 2, the analytical solution is plotted as a blue line and the red x-marks are the determined values.

Figure 2 only gives information of the chosen points, more points means that a more accurate comparison can be made. In this case 28 points are chosen to be analyzed. What can be noticed directly is that the x-marks do not begin and end at the relative length of zero and 1 respectively.

This is because the Matlab script uses the half-way bounce back method and this method gives only values between grid nodes.

What can be concluded from the 28 points is that their values are correspondent to the analytical solution. The conclusion from this result is that, in the case of the Poiseuille flow, the Matlab script gives logical values and represents the reality. However, with only one benchmark the reliability of the result is still questionable; hence a second benchmark is carried out.

4.2.2 Flow around the Cross-Section of a Cylinder

The second benchmark is a more complex process to calculate the flow of RBCs analytically, as well as to compute in Matlab. It is the flow around the cross-section of a cylinder which is centered in the middle of the grid without solid walls. This investigation is to analyze the effect (through the Matlab program) of the flow of RBCs in the plasma layer near the arteriole wall on nitric oxide (NO) and oxygen (O₂) transport.

First the analytical solution will be given, after which the results will be compared with the results of the Matlab script.

4.2.2.1 The Analytical Solution

Whenever an infinitely long cylinder, with radius a , is moving through an infinite stationary fluid with a constant velocity U , one can solve the Navier-Stokes equation to obtain the velocity profile. Using the Stokes stream function ψ , the problem of solving the Navier-Stokes equation is reduced to solving a partial differential equation for ψ . The problem is simplified even more by choosing the proper boundary conditions. This will reduce the problem to that of solving an ordinary differential equation in order to obtain ψ .

4.2.2.2 Solving the Navier-Stokes Equation

The Navier-Stokes equation for an incompressible fluid is given by [15] and in equation (4.1)

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho(\vec{u} \cdot \nabla \vec{u}) = -\nabla \vec{p} + \mu \nabla^2 \vec{u} + \vec{F}_{ext} \quad 4.6$$

in which ρ is the density of the fluid, $\vec{u}$ is the velocity of the fluid, p is the pressure, μ is the dynamic viscosity of the fluid and $\vec{F}_{ext}$ is a term which is nonzero when there are external body forces. The assumption that the Reynolds number is very low results in a reduction of the Navier-Stokes equation due to the fact that for low Reynolds numbers the term $(\vec{u} \cdot \nabla) \vec{u}$ may be disregarded. The equation is simplified further because a stationary solution has to be found, [8]. The simplified version of the Navier-Stokes equation for low Reynolds number is

$$\nabla p = \mu \nabla^2 \vec{u} \quad 4.7$$

The rule of Laplace states that the Laplace operator can be rewritten as

$$\nabla^2 \vec{u} = \nabla \times (\nabla \times \vec{u}) - \nabla(\nabla \cdot \vec{u}) \quad 4.8$$

Combining equation (4.8) and the continuity condition which says that $\nabla \cdot \vec{u} = 0$, equation (4.7) can be reduced to

$$\nabla p = -\mu \nabla \times (\nabla \times \vec{u}) = -\mu \nabla \times \vec{\Omega} \quad 4.9$$

in which the vector $\vec{\Omega}$ has been introduced. The vector is given by $\vec{\Omega} = \nabla \times \vec{u}$ because the problem is symmetrical in the axis, an axisymmetric velocity field will be sought in the form of

$$\vec{u} = u_r(r, \theta) \hat{e}_r + u_\theta(r, \theta) \hat{e}_\theta \quad 4.10$$

4.2.2.3 Introducing the Stokes Stream Function

The divergence $\nabla \cdot \vec{u}$ for an axisymmetric velocity field in spherical coordinates is given by

$$\nabla \cdot \vec{u} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (u_\theta \sin \theta) \quad 4.11$$

Using the continuity equation and the introduction of the Stokes stream function $\psi(r, \theta)$ will give an expression for u_r and u_θ

$$u_r = \frac{1}{r^2 \sin \theta} \frac{\partial \psi}{\partial \theta} \quad 4.12a$$

$$u_{\theta} = \frac{-1}{r \sin \theta} \frac{\partial \psi}{\partial \theta}$$

4.12b

Equations (4.12) show that the velocity profile is only dependent on the stream function. In order to determine the stream function from the Navier-Stokes equation one has to calculate

$$\vec{\Omega} = \nabla \times \vec{u} = \Omega_r \hat{e}_\theta + \Omega_\theta \hat{e}_\varphi \quad 4.13$$

Since $\vec{u}$ is axisymmetric the r and θ components are zero leaving only the φ component.

$$\begin{aligned} \Omega_\varphi &= \frac{1}{r} \left(\frac{\partial}{\partial r} (r u_\theta) - \frac{\partial u_r}{\partial \theta} \right) \\ &= \frac{1}{r} \left[\left(\frac{\partial}{\partial r} \left(\frac{-1}{\sin \theta} \frac{\partial \psi}{\partial r} \right) - \frac{\partial}{\partial \theta} \left(\frac{1}{r^2 \sin \theta} \frac{\partial \psi}{\partial \theta} \right) \right) \right] \\ &= \frac{-1}{\sin \theta} \left[\left(\frac{\partial^2 \psi}{\partial r^2} + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial \psi}{\partial \theta} \right) \right) \right] \\ &= \frac{-1}{r \sin \theta} E^2 \psi \end{aligned} \quad 4.14$$

in which E^2 is a differential operator and is defined as

$$E^2 = \frac{\partial^2}{\partial r^2} + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial \psi}{\partial \theta} \right) \quad 4.15$$

Inserting this equation in equation (4.9) results in

$$\begin{aligned} \nabla p &= -\mu \nabla \times (\nabla \times \vec{u}) = -\mu \nabla \times \vec{\Omega} \\ &= \frac{-\mu}{r \sin \theta} \left(\frac{\partial}{\partial \theta} \left(\frac{-1}{r} E^2 \psi \right) \right) \hat{e}_r + \frac{\mu}{r} \left(\frac{\partial}{\partial r} \left(\frac{-1}{\sin \theta} E^2 \psi \right) \right) \hat{e}_\theta \\ &= \frac{-\mu}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (E^2 \psi) \hat{e}_r - \frac{\mu}{r \sin \theta} \left(\frac{\partial}{\partial r} (E^2 \psi) \right) \hat{e}_\theta \end{aligned} \quad 4.16$$

Due to the fact that $\nabla p = \frac{\partial p}{\partial r} \hat{e}_r + \frac{1}{r} \frac{\partial p}{\partial \theta} \hat{e}_\theta + \frac{1}{r \sin \theta} \frac{\partial p}{\partial \varphi} \hat{e}_\varphi$ one can obtain two differential equations for the pressure [8]

$$\frac{\partial p}{\partial r} = \frac{\mu}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (E^2 \psi) \quad 4.17a$$

$$\frac{\partial p}{\partial \theta} = -\frac{\mu}{r \sin \theta} \left(\frac{\partial}{\partial r} (E^2 \psi) \right) \quad 4.17b$$

When these two equations are cross-differentiated they become

$$\frac{\partial}{\partial \theta} \left(\frac{\partial p}{\partial r} \right) = \frac{\mu}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} (E^2 \psi) \right) \quad 4.18a$$

$$\frac{\partial}{\partial r} \left(\frac{\partial p}{\partial \theta} \right) = -\frac{\mu}{\sin \theta} \frac{\partial^2}{\partial r^2} (E^2 \psi) \quad 4.18b$$

These two cross-differentiations must be equal, resulting in

$$\frac{\partial^2}{\partial r^2} (E^2 \psi) + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} (E^2 \psi) \right) = E^2 (E^2 \psi) = 0 \quad 4.19$$

Inserting the formula for the differential operator E^2 in equation (4.19) yields

$$\left[\frac{\partial^2}{\partial r^2} + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \right) \right]^2 \psi = 0 \quad 4.20$$

Solving this equation requires boundary conditions. For this problem two boundary conditions are imposed. The first is the no-slip condition on the surface of the cylinder resulting in $\vec{u}(a, \theta) = 0$. The second boundary condition says that the flow far from the cylinder has a constant velocity U . The no-slip condition implies that

$$\frac{\partial \psi}{\partial r} \Big|_{r=a} = 0 \quad 4.21a$$

$$\frac{\partial \psi}{\partial \theta} \Big|_{r=a} = 0 \quad 4.21b$$

The second boundary condition can be formulated using limits. The second boundary condition is

$$\lim_{r \rightarrow \infty} u_r = U \cos \theta \quad 4.22a$$

$$\lim_{r \rightarrow \infty} u_\theta = -U \cos \theta \quad 4.22b$$

This boundary condition needs to be reformulated so it will be a requirement for ψ instead of $\tilde{u}$. Using equations (4.12) the second boundary conditions can be rewritten and one can easily find the new boundary condition for ψ

$$\lim_{r \rightarrow \infty} \psi = \frac{1}{2} U^2 \sin^2 \theta \quad 4.23$$

This boundary condition results in a prediction of how the final stream function will look like. The expectation is that the stream function should have the following form

$$\psi = f(r) \sin^2 \theta \quad 4.24$$

To check this solution the differential operator E^2 is applied to equation (2.18)

$$\begin{aligned} E^2(f(r) \sin^2 \theta) &= \frac{\partial^2}{\partial r^2} (f(r) \sin^2 \theta) + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} (f(r) \sin^2 \theta) \right) \\ &= \frac{\partial^2 f(r)}{\partial r^2} \sin^2 \theta + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left(\frac{f(r)}{\sin \theta} 2 \sin \theta \cos \theta \right) \\ &= \left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right) f(r) \sin^2 \theta \end{aligned} \quad 4.25$$

To simplify the equation a dummy function is introduced, represented by

$$g(r) = \left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right) f(r) \quad 4.26$$

After introducing this dummy function the differential operator E^2 is again applied to equation (4.25), resulting in

$$\begin{aligned} E^2 &= (E^2(f(r) \sin^2 \theta)) = E^2(g(r) \sin^2 \theta) \\ &= \left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right) g(r) \sin^2 \theta \\ &= \left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right)^2 f(r) \sin^2 \theta \end{aligned} \quad 4.27$$

If the trial function given by equation (4.24) satisfies the differential equation for ψ , which is given by equation (4.19), then equation (4.27) must be equal to zero. This results in a requirement for $f(r)$ in the form of an ordinary differential equation

$$\left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right)^2 f(r) = 0 \quad 4.28$$

The form of the equation implies a solution of the type $(r) = r^\alpha$. Inserting this into equation (4.28) results in

$$\left(\frac{d^2}{dr^2} - \frac{2}{r^2} \right)^2 r^\alpha = 0 \quad 4.29$$

And one can easily find the solution α

$$(\alpha(\alpha-1)-2)((\alpha-2)(\alpha-3)-2) = 0 \quad 4.30$$

The solutions of equation (4.30) are $\alpha = -1, 1, 2$ and 4 . Now these solutions can be implemented in the solution to the partial differential equation for ψ in the form of

$$\psi = (Ar^{-1} + Br^1 + Cr^2 + Dr^4) \sin \theta \quad 4.31$$

The boundary conditions can be used to find the variables A, B, C and D . The second boundary condition (uniform flow far from the cylinder) requires that Cr^2 is the dominant term as $r \rightarrow \infty$ which in turn implies $D = 0$. Using equation (4.23) one can find that

$$C = \frac{1}{2} U \quad 4.32$$

The no-slip condition leads to two requirements of the derivatives of ψ . These are used to determine the two remaining constants, A and B . Using equations (4.21) results in two functions

$$-\frac{A}{a^2} + B + Ua = 0 \quad 4.33a$$

$$\frac{A}{a} + Ba + \frac{1}{2} Ua^2 = 0 \quad 4.33b$$

These equations form a set which can be solved using linear algebra. Using this technique one can find that

$$A = \frac{1}{4} Ua^3 \quad 4.34a$$

$$B = -\frac{3}{4}Ua$$

4.34b

Now that the variables A , B , C and D are known, the stream function can be obtained

$$\psi = \frac{1}{4}U(2r^2 - 3ar + a^3r^{-1})\sin^2\theta \quad 4.35$$

Combining equations (4.12) and equation (4.35), Versteeg, H. K., & Malalasekera, W. (2007), the velocity profile of the fluid can be found

$$\vec{u} = U\cos\theta \left(1 - \frac{3a}{2r} + \frac{a^3}{2r^3}\right)\hat{e}_r - U\sin\theta \left(1 - \frac{3a}{4r} + \frac{a^3}{4r^3}\right)\hat{e}_\theta \quad 4.36$$

4.2.2.4 Results of the Analytical Flow Around the Cross-Section of a Cylinder

To visualize the analytically calculated flow around the cross-section of a cylinder given by equation (4.36), a Matlab program has been written to simulate the flow. This program makes a surface plot of the velocity field as shown in figure 3.

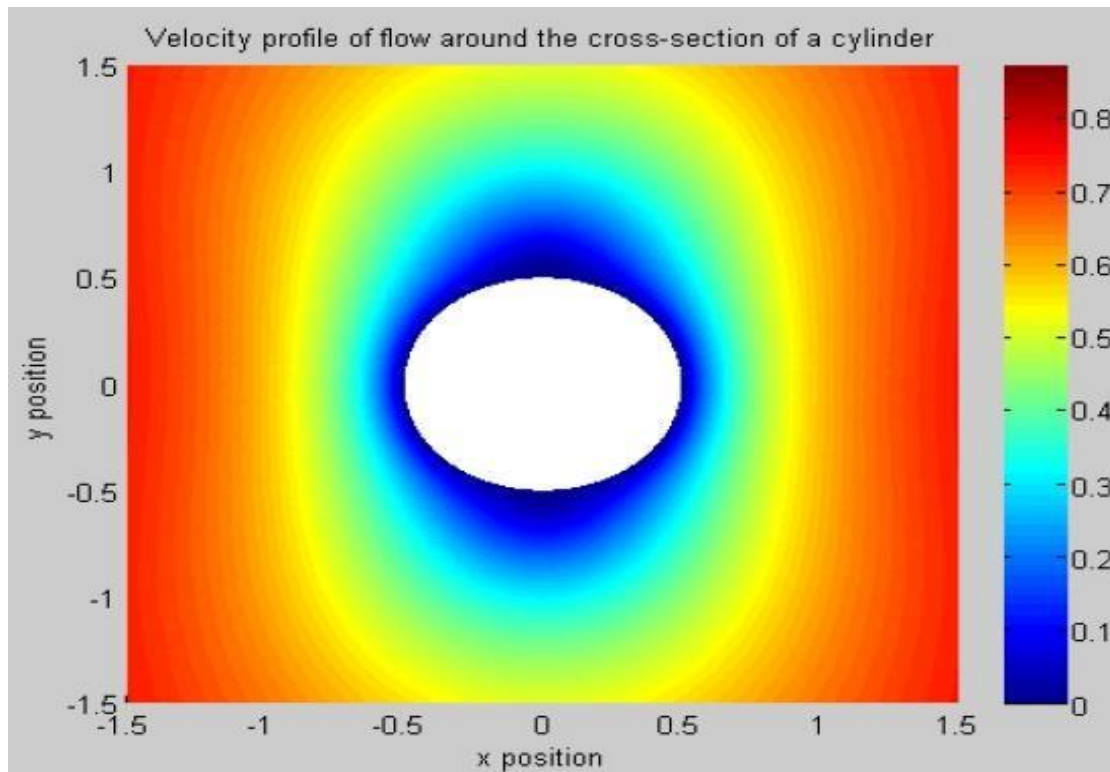


Figure 3: The surface plot of the analytically determined velocity field of the flow around the cross-section of a cylinder

4.2.2.5 The Computationally Determined Solution

For the Matlab script to give the flow around the cross-section of a cylinder, the boundaries are set properly. The boundaries are built by hand in a high precision model, which represents the cross-section of a cylinder good while keeping the computational time limited.

4.2.2.5.1 Results of the computationally determined flow around the cross-section of a cylinder

The Matlab program has been modified in order to give the velocity field around a model of the cross-section of a cylinder. This result is shown in figure 4.

4.2.2.6 Comparing the Two Results

Comparing figure 3 and figure 4 gives an interesting result. One can see that inside the cross-sections the velocity in both figures is zero. Also the velocity field far away from the cross-section of the cylinder becomes uniform which is shown in both figures. In figure 4.4 however, the full axes are not shown. Because of this the flow near the sides of the figure are not uniform and could therefore be misinterpreted. The major similarity is that the flow around both cross-sections is similar i.e (figure 3 has a real sphere and figure 4 has a modeled sphere), the velocity becomes slower and follows the boundaries of the cross-sections smoothly.

The only difference between both figures is that in figure 4, the cross-section is not smooth resulting in velocity vectors at places where there should not be any. However, this is not the result of a mistake in the Matlab program but it is due to lack of accuracy of the

model of the cross-section. Figure 4 has a high degree of accuracy making it safe to say that both figures should become similar when the accuracy of figure 4 increases even more. However, this is not done due to large computational time needed to perform such calculations. Another difference is that the domain which is used is not infinitely large, which will result in different outcomes than the analytical solution.

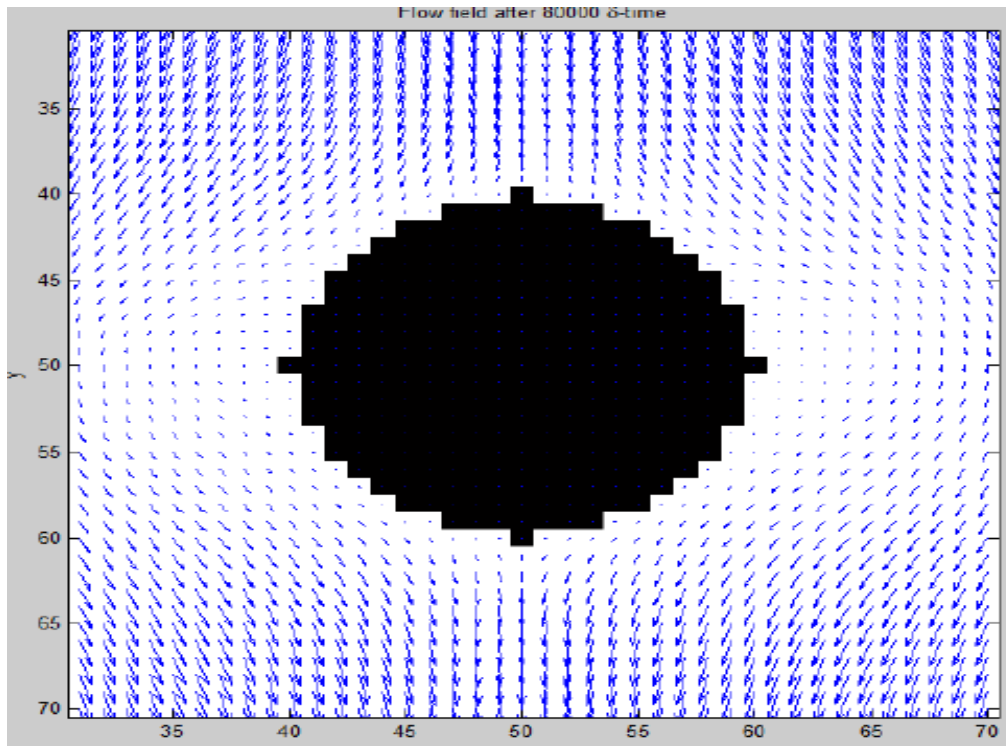


Figure 4: The computationally determined velocity field around a model of the cross-section of a cylinder

4.3 Final Conclusion of the Benchmarks

In both benchmarks, the Matlab program performed as expected. The first benchmark was the Poiseuille flow and due to all the similarities and the absence of large differences, the Matlab program for this case can be considered to be correct. The second benchmark was a more complex one, not only to solve analytically but it took also a lot more computational time (e.g. the computational time needed for a 100x100 grid was 34 hours on a regular laptop). However, because of all the similarities and the only difference caused by lack of accuracy it is again safe to say that the Matlab program gives the correct answers. The final conclusion of the benchmarks is that in both cases the Matlab program gives physically plausible results which are the same as the analytically determined results. The Matlab program can therefore be considered to be correct.

4.4 Determining the Error of the Used Lattice Boltzmann Method

In order to get an indication of the accuracy of the used method, the error is determined. To determine the error, four resolutions have been made of a cross-section of a cylinder, each having an increase in the accuracy. For these four resolutions several points in the plot have been compared with the results of the analytically determined values of those points, after which an error has been calculated using equation (4.37) [1]

$$\varepsilon = \frac{|value_{analytical} - value_{numerical}|}{value_{analytical}} \quad 4.37$$

The errors of the different models have been calculated using a program made in excel. The results of the excel program were imported in Matlab where further calculations were made [15]. In this section, the outcome of the determination of the error of the used method is shown.

4.5 Result

In both benchmark cases the results are plausible, physically correct and have many similarities with the analytical results. Therefore the Matlab program is considered to be correct, meaning that more complex geometries can be evaluated and the results can also be considered to be correct. The results of the determination of the error of the used method are shown first, after which the results of the analysis of the stacking pattern are shown.

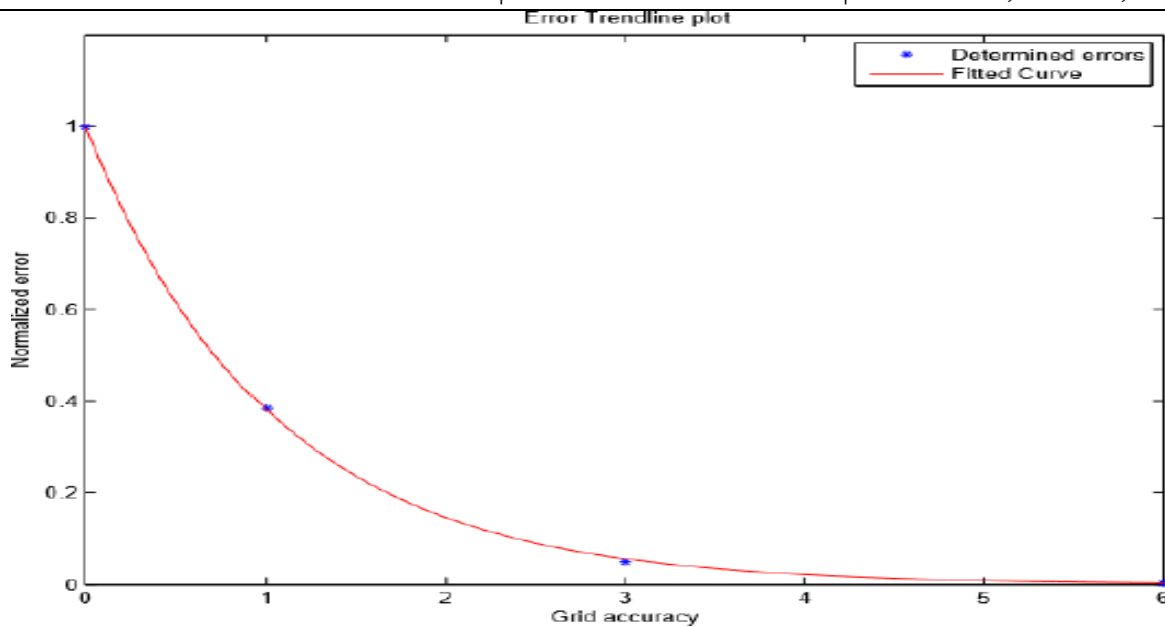


Figure 5: The plot of the normalized error vs the grid accuracy of the flow around the crosssection of a cylinder on regular axes

4.5.1 The Determination of the Error of the Used Method

The error in the method used has been determined using four resolutions of increasing accuracy (i.e. more grid nodes used to tighten the grid). The errors were determined using an excel program after which Matlab was used to make plots of the error vs the grid accuracy. Two plots were made as shown in figures 5 and 6, one on regular axes and one on semi-log axes (i.e. the y-axis is logarithmic).

Figure 5, shows that when the accuracy of the grid increases the normalized error drops like an exponential function. To confirm this estimation another plot is made in Matlab using semi-logaxes. This means that the y-axis has a logarithmic scale and the x-axis has a regular scale. The results of this semi-log plot are shown in figure 6.

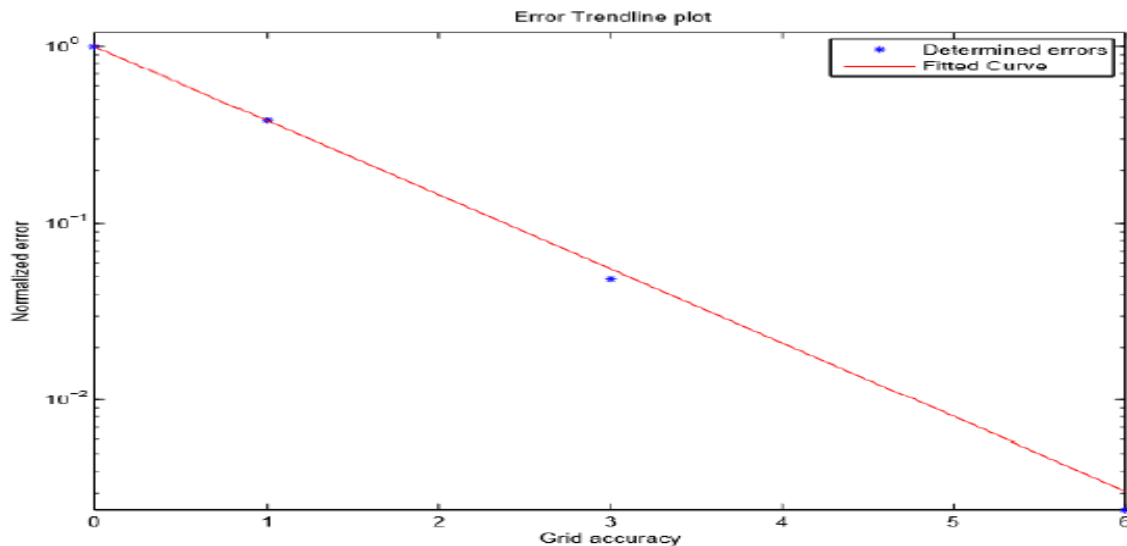


Figure 6: The plot of the normalized error vs the grid accuracy on semi-log axes

These plots are used in order to get a better understanding of the order of the error. This is done by using the "fit" function in Matlab. The function gives the parameters of the estimated function which connects the points in the best way possible (i.e. Matlab uses the least squares method). From figure 4.5 and 4.6, the function is estimated to be an exponential function of the form

$$fit = A(Bx) \quad 4.38$$

In which A and B are constants with A = 1 since it is normalized and B = -0.9639, which is estimated by Matlab with an accuracy of 95%. This results in the function of the error as

$$error = \exp(-0.9639x) \quad 4.39$$

In equation (4.39), x represents the accuracy of the grid which is the inverse of the error of the grid. From equation (4.39) it can be seen

that the chosen method is of the first order. Since the half-way bounce back method is used, this result corresponds with the one mentioned in the literature

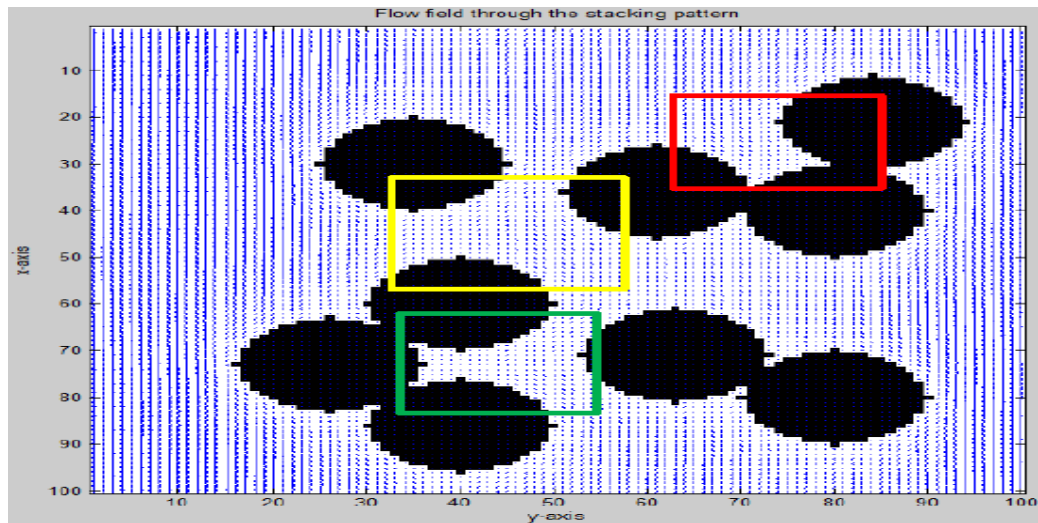


Figure 8: The complete view of the vector field created by Matlab with the marked areas which will be looked at more with more detail (i.e. red = area 1, green = area 2, yellow = area 3). Parameters used are $\tau = 0.6$, $Re < 100$, $D = 20$ lu

Figure 4.7: The stacking pattern which is analyzed in Matlab using the lattice Boltzmann method

After running the Matlab program a vector field is created which shows how the fluid moves through this specific stacking pattern. However, since the grid has a high accuracy the vector field can not be analyzed when looking at the complete picture as one can see in figure 8.

To analyze the vector field, three areas are looked upon more closely. The first area (marked red in figure 8) which is analyzed is the cluster of three cross-sections of cylinders in the top-right of figure 8. The second area (marked green in figure 8) is the other cluster of three cross-sections of cylinders in the bottom-left corner. These two areas both have a cluster of three cross-sections but they are oriented in different ways. The last area (marked yellow in figure 8) which is analyzed is the pathway through the middle of the cross-sections. These areas are chosen because in these areas the most interesting things will happen.

4.5.2.1 Analyzing area 1

The first area which will be analyzed is the one marked red in figure 8. This area is chosen because in the cluster of the cross-sections there is a gap, where the fluid will come to a dead-end. As shown in figure 9 the incoming fluid moves around the cross-sections towards the gap. Once the fluid is in the gap the velocity drops rapidly and the fluid stands almost completely still at the end of the gap. From figure 9 it can be seen that there is a mass accumulating in the gap, which is not physically possible.

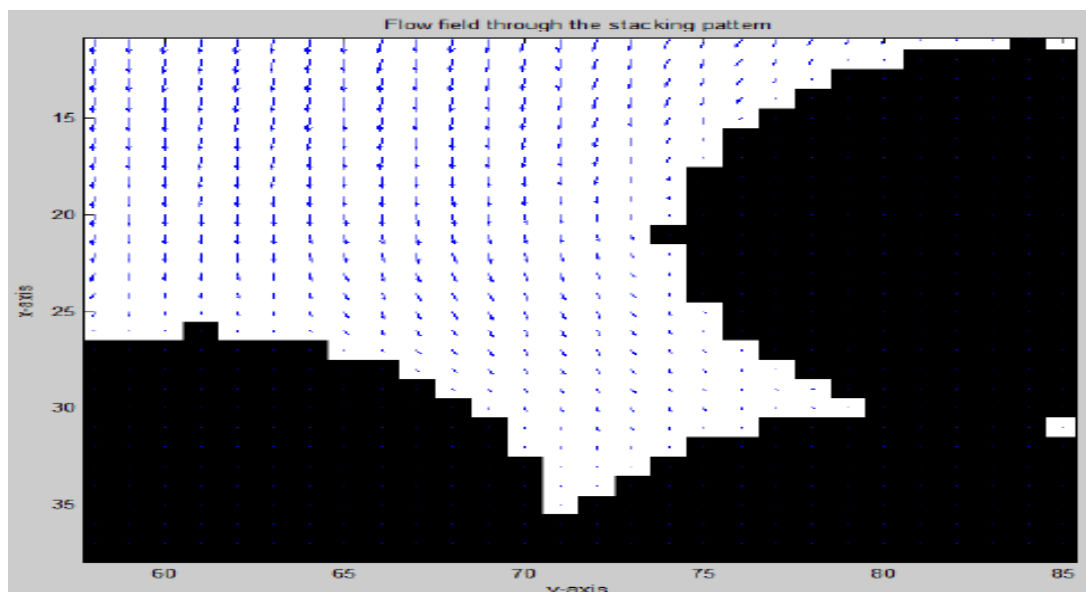


Figure 9: The vector field of the analyzed stacking pattern focused on area 1

4.5.2.2 Analyzing area 2

The second area which will be analyzed is a similar one to area 1 but the orientation of the gap is now to the east side instead of to the north side. Another difference is that the incoming fluid is squeezed through a narrow path as one can see in figure 8. The vector field of area 2 is shown in figure 10. What can be observed is that, due to the orientation of the gap, the fluid in the gap almost stands still. This resistance is much greater in the gap than through the pathway, therefore the fluid has no velocity in the large portion of the gap.

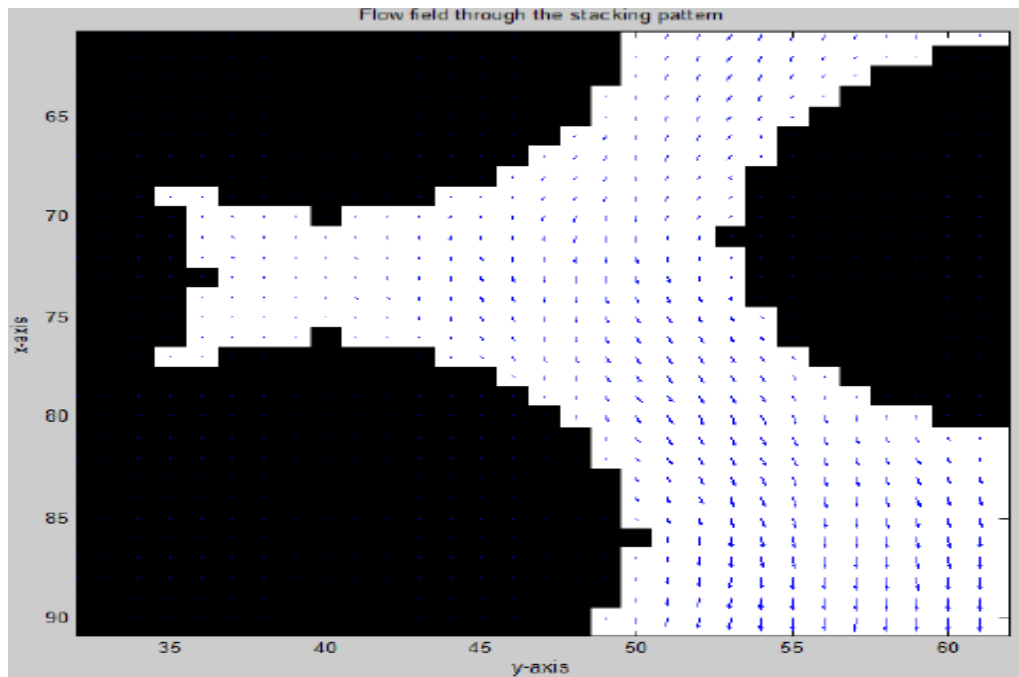


Figure 10: The vector field of the analyzed stacking pattern focused on area 2

4.5.2.3 Analyzing area 3

The last area which is analyzed is the one marked yellow in figure 8. It is the area between the majorities of the cross-sections in which the most preferable pathway is examined. In figure 8 the most preferable pathway can be seen when looking at the density of the vectors, but a more detailed overview can be found in figure 11. From figure 11 the most preferable pathway can be noticed. It is the path passing by the right side of the bottom-left cluster. This is in accordance with the expectations, due to expected lower resistance in the right side of the bottom-left cluster in comparison with the left side of this cluster.

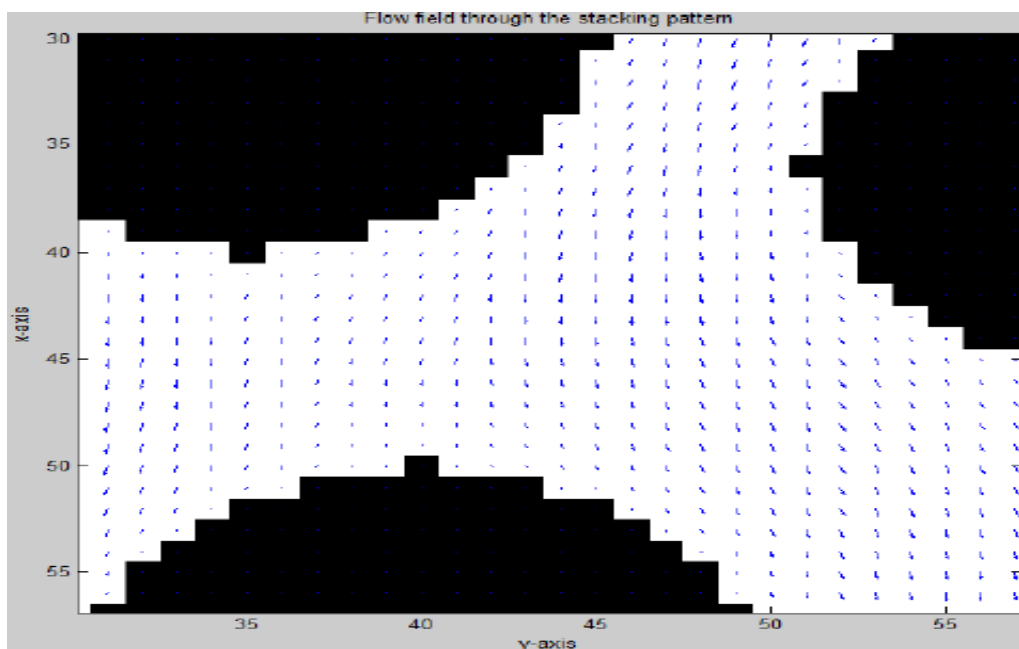


Figure 11: The vector field of the analyzed stacking pattern focused on area 3

5.1 Summary

RBCs flow and its behavior was modeled and analyzed using the D2Q9 lattice Boltzmann method created in Matlab. Prior to the analysis of the determined values, the Matlab script was benchmarked using the Poiseuille flow and the flow around the cross-section of a cylinder, after which the accuracy of the used method was determined.

5.2 Conclusion

The conclusions for the vector field are based upon the results and expectations due to fluid resistance.

The first benchmark is the Poiseuille flow. The analytically determined values resulted in a graph, which can be seen in figure 1. Using the created Matlab script the velocities of 28 points were calculated. These values were plotted in the same graph as the analytically determined graph.

The second benchmark is the flow around a cross-section of a cylinder. The analytical derivation is more complex than the Poiseuille flow but it resulted in a surface plot, which can be seen in figure 2. After the result of the analytically determined flow field was known, the Matlab script was used in order to get a high precision model of a cross-section of a cylinder, which was used to model the fluid flow. Due to the similarities between the modeled flow field and the analytically determined flow field, the Matlab script is considered to be correct regarding the second benchmark.

The Matlab code is used in modelling the cross-section of a cylinder. Since it is a model the accuracy of the used model was examined. The accuracy was determined using a created excel program combined with Matlab. The accuracy of the modeled cross-section was enlarged and with each enlargement, the error was shown between the obtained data and the analytically determined data. Finally a plot was made of the error vs the accuracy, which can be seen in figures 5 and 6. Using Matlab's internal function an estimate was made of the order of the error of the used lattice Boltzmann model. The result is that the error was of the first order which is in accordance with the literature value for the half-way bounce back method.

A certain geometry presented in figure 7, was created and analyzed using three areas in which more detail was visible.

The first area resulted in a vector field towards a cluster of cross-sections with a gap positioned north, which can be seen in figure 9. The fluid flows in the gap and the fluid velocity goes to zero as the fluid comes closer to the boundaries. This is in accordance with the expectations because the fluid comes to a dead-end from which it cannot escape resulting in a very small velocity in the gap. Hence the analysis revealed that the stiffness of the cell membrane and the viscosity of the intracellular fluid relative to that of surrounding fluid, which is the plasma, can greatly affect the deformation and motion of the RBC.

The second area is similar to the first one but the orientation of the gap of this cluster is east ward and a different entry of the fluid. This resulted in an almost stagnant flow in the gap due to the higher fluid resistance in the gap in comparison with the pathway between the boundaries. This is also in accordance with the expectations, because the gap creates a large fluid resistance and the fluid flows with a smaller resistance through the pathway resulting in a preferable path the fluid will take, which is visible in figure 10. This also reveals that when a particular pathway in the arteries is block or narrowly closed, the flow of the RBCs or blood in its entirety may experience a higher fluid resistance due to the opening, and possibly take alternative path if there is any. The block arteries in the human body may cause the body system to experience a very poor regulation in the microcirculation, transportation of oxygen and nutrients, and the body's immunological and inflammatory responses.

The last area is the area between the majorities of the cross-sections in which the most preferable path is examined. This path runs through the middle to the right side of the bottom-left cluster, which can be seen in figure 11. This is also in accordance with the expectations, because the resistance the fluid encounters is expected to be higher to the left of the cluster than to the right side. This is due to the small asymmetry between the placements of the three cross-sections visible in area 3. Due to this asymmetry the fluid is expected to follow the path to the right side of the bottom-left cluster. This expectation is in accordance with the obtained result from Matlab.

Finally, the formulated script have provided a simulation technique employed in modeling the flow of red blood cells (RBCs) in blood plasma or arteries using a matlab program which also described the entire and possible analysis.

5.3 Contribution to Knowledge

From the simulation and analysis, it is a possible fact that LBM can be modeled analytically and a matlab program can be used to formulate codes which is capable of creating vector plots representing the RBCs flow through a stacking pattern and this demonstrates the possible behavior of the fluid when subjected to a poiseuille flow and a flow around a cross-section using cylindrical approach to represent the flow behavior of the RBCs in the cardiovascular system or arteries.

5.4 Recommendation

First, my recommendation is centered on the Matlab programs formulated. From the benchmarks the computational program can be considered to give similar results in comparison with the analytically determined results. This can be tried in 3 dimensional space. However, this remains uncertain and in order to be sure what is wrong more stacking patterns have to be analyzed and further improvements have to be made in the Matlab program. Since the mass accumulation error is the only mistake that can be noticed from

the vector field of the stacking pattern the Matlab program is not useless, it still gives logical overall fluid fields, but in order to be completely satisfied further research is needed in order to solve the mass accumulation problem.

Very importantly, to strengthen the research, it is highly recommended to extend the analysis beyond a single RBC stacking pattern and incorporate a comparative study between the D2Q9 lattice Boltzmann method (LBM) and commercial CFD software like CFX or ANSYS. This comparison will provide valuable insights into the strengths, limitations, and potential biases of each approach, ultimately enhancing the reliability and generalizability of the findings.

However, as with any numerical approach, the topic of computational efficiency and fidelity ought to be considered. This sensitivity is particularly pertinent to the study of RBC flow behaviour which feature sizes such as RBC diameters which are typically on the micrometer scale. Correspondingly, the scale of discretisation for the numerical model can be on the order of nanometres in order to preserve the accuracy and fidelity of the simulation. This is problematic for studies that are essentially multiscale in nature, such as the study of a capillary network or an organ. The modelled domain in its entirety is in several order larger than the discretisation scale required to capture reasonably correct flow physics. Understandably, the computational cost for such studies will be high. Therefore, strategies such as parallel computing techniques for large multiscale RBC simulations need to be developed in order to study the many practical biological systems. From the perspective of applied models, many studies are trending towards tackling large multiscale problems. Furthermore, the popularity of multicore computing provides a huge potential for more efficient computational algorithms for solving mathematical RBC models numerically.

5.5 Conflicts of Interest

The authors declare that there are no conflicts of interest relating to the content or publication of this paper.

References

1. Chung, T. J. (2010). Computational Fluid Dynamics. In T. J. Chung, Computational Fluid Dynamics. Cambridge: Cambridge University Press.
2. Chen, S., & Doolen, G. D. (1998). Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1), 329-364.
3. Dadvand A, Baghalnezhad M, Mirzaee I, Khoo B. C, and Ghoreishi S. (2014). "An immersed boundary-lattice Boltzmann approach to study the dynamics of elastic membranes in viscous shear flows," *Journal of Computational Science*, vol. 5, no. 5, pp. 709–718.
4. Hardy, J., & de Pazzis, Y. (1976). Models of lattice gas automata for simulating fluid dynamics. *Journal of Statistical Physics*, 15(6), 129-148.
5. Hardy, J., and de Pazzis, O. (1976). Molecular dynamics of a classical lattice gas. In J. Hardy, & O. de Pazzis, *Molecular dynamics of a classical lattice gas*.
6. Hess, J. L. (1967). Calculation of Potential Flow About Arbitrary Bodies. In J. L. Hess, *Calculation of Potential Flow About Arbitrary Bodies*.
7. Hess, J., & Smith, A. M. O. (1967). Calculation of potential flow about arbitrary bodies. *Journal of Ship Research*, 9(2), 22-44.
8. Kundu, P. K., Cohen, I. M., & Dowling, D. R. (2015). *Fluid Mechanics* (6th ed.). Academic Press.
9. Mohamad, A. A. (2011). *Lattice Boltzmann method: Fundamentals and engineering applications with computer codes*. Springer.
10. Niclas Berg (2018) *Blood flow and cell transport in arteries and medical assist devices*. ISBN: 978-91-7873-037-7
11. Rohde, M. (2004). Extending the lattice Boltzmann method. Delft.
12. Rohde, C. (2009). *Lattice-Boltzmann and finite-difference lattice-Boltzmann methods for fluid dynamics*. Elsevier.
13. Rohde, M. (2009). Cellular Automata. In M. Rohde, *Cellular Automata*. Delft. walls at the same scale," *Acta Physica Sinica*, vol. 63, article 174701, no. 17, 2014 (Chinese).
14. Skalak, R., Tozeren, A., Zarda, R. P., & Chien, S. (1973). Biophysical strain energy function of red blood cell membranes. *Biophysical Journal*, 13(9), 245-264.
15. Versteeg, H. K., & Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method* (2nd ed.). Pearson Education.