

Bachelor thesis

Mikrofluidic elements

Ilia Fedotov

15.01.2025



Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Institut für Mikrosystemtechnik - IMTEK

Eingereichte Bachelorarbeit gemäß den Bestimmungen der Prüfungsordnung der Albert-Ludwig-Universität Freiburg für den Studiengang Bachelor of Science (B. Sc.) Embedded Systems Engineering vom 2018.

Bearbeitungszeit:

15.10.2024 bis 15.01.2025

Gutachter:

Dr. Andreas Greiner

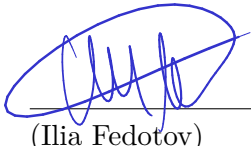
Betreuer:

Dr. Andreas Greiner, Johannes L. Hörmann

Erklärung zur Selbstständigkeit

Hiermit versichere ich, die vorliegende Dissertation selbständig und nur mit Hilfe der angegebenen Quellen und Hilfsmittel angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sowie alle Daten, die aus Kollaborationen stammen, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen, ein Promotionsversuch wurde von mir bisher nicht unternommen.

Freiburg, 15.01.2025



(Ilia Fedotov)

Abstract

This study aims to simulate the results of the incompressible Navier-Stokes Equations for flow limitation due to an obstacle in a narrow channel and compare them to the findings in a recent study. We used numerical simulation techniques, utilizing DOLFINx and the Finite Element Method to solve the Partial Differential Equation. Our key findings reveal that for the incompressible Navier-Stokes Equations, the results align with expectations in the static case and show plausible results for iterative shape changes using a string deformation ordinary differential equation.

Kurzfassung

In unserer studie simulieren wir die Ergebnisse der inkompressiblen Navier-Stokes-Gleichungen(N-S) für Strömungsbegrenzung und vergleicht sie mit den Erkenntnissen einer kürzlich durchgeführten studie. Wir verwendeten numerische Simulationsmethoden mit DOLFINx [1] und der Finite-Elemente-Methode(FEM) zur Lösung der partiellen Differentialgleichungen. In unsren Simulationen konnten wir zeigen, dass für die inkompressiblen N-S im statischen Fall die Ergebnisse den Erwartungen entsprechen und plausible Ergebnisse für die iterativen Membranänderung (Strömungsbegrenzung) durch einer gewöhnlichen Differentialgleichung zur Saitenverformung liefern.

Glossary

ALU Albert-Ludwigs-Universität.

cfl Courant-Friedrichs-Lewy condition.

DOLFINx DOLFINx.

FEM Finite element method.

IMTEK Institut für Mikrosystemtechnik.

IPCS Incremental Pressure Correction Scheme.

lhs left hand side.

N-S Navier Stokes Equation.

ODE Ordinary Differential Equation.

PDE Partial Differential Equation.

PDF Portable Document Format.

rhs right hand side.

Symbols

S Function representing a slice through/the coordinates of a domain.

c courant number.

P_0 dimensionless pressure at the end of the membrane.

P_G dimensionless pressure at the membrane or gate.

P_L dimensionless pressure at x.

P_{SD} dimensionless pressure gradient from inlet to outlet respectively source to drain.

Q Mass flow rate.

T String stiffness.

Ω Domain.

\bar{p} dimensionless p.

\bar{t} dimensionless t.

\bar{u} dimensionless u.

\bar{x} dimensionless x.

\bar{y} dimensionless y.

\bar{z} dimensionless z.

Re Reynolds number.

μ dynamic viscosity.

ν Poisson rate.

ρ density.

σ linear elasticity.

ζ equilibrium pressure equation between P_{SD} and P_G based on a string equation.

p pressure.

Table of Contents

Abstract	i
Kurzfassung	ii
Glossary & Symbole	iii
1 Introduction	1
1.1 Motivation	1
1.2 Identification of the Scientific Question and needed Investigation	1
2 Theory	3
2.1 Governing Equations System	3
2.2 Equations	3
2.2.1 General	3
2.2.2 Assumptions	4
2.2.3 Analytical solutions	4
2.2.4 Introduction of Dimensionless variables	5
2.2.5 Incremental Pressure Correction Scheme	5
2.2.6 Deflection of a membrane in a channel	6
2.3 Simulation	7
2.3.1 Stability - Courant-Friedrichs-Lewy condition(CFL)	7
2.3.2 Units	8
2.3.3 Mesh	8
2.3.4 Boundary conditions	8
2.3.5 DOLFINx	9
2.4 Dtool	11
3 Results and Discussion	12
3.1 Results	12
3.1.1 Finding suitable parameter dimensions	12
3.1.2 Parametric study with half circle as obstacle	13
3.1.3 Parametric study with iterative membrane as obstacle	15
3.2 Discussion	18
3.2.1 Mass flow rate	20
3.2.2 Pressure	21
3.2.3 Velocity	22

Table of Contents

4 Summary and Conclusion	24
Acknowledgements	27
References	28

List of Figures

1.1	Part of a generated mesh of a canal with restricted flow showing top and bottom boundary conditions and indicating directions of in- and outflow.	2
2.1	Rectangular Channel	7
3.1	Mass flow rate for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.	13
3.2	Mass flow rate in the center of canal over pressure difference at 0.6.	14
3.3	Pressure for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.	15
3.4	Velocity for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.	15
3.5	Membrane (a) shapes and (b) velocities for parametric membrane simulation.	16
3.6	Change in mass flow rate and velocity for different P_G and membranes. (a) The change in mass flow rate. (b) The change of velocity.	17
3.7	Pressure over membrane length of 2 at 0.32 with P_{SD} of 550.	17
3.8	Pressure in the center of the canal at 0.32.	18
3.9	Mass flow rate in the center of canal over pressure difference at 0.32.	18
3.10	Analytical mass flow rate for different P_{SD} compared to simulated.	19
3.11	Analytical velocity for different P_{SD} compared to simulated.	20
3.12	Change in mass flow rate for different P_G and radii compared to different mass flow rates of the membrane simulation. (a) Change in mass flow rate for different P_G and radii compared to different mass flow rates of the membrane simulation. (b) The change of mass flow rate for different P_G and membranes.	21
3.13	Comparing pressure from the second and third simulation. (a) Pressure at the center of the canal at 0.6 for radial obstacles. (b) Pressure at the center of the canal at 0.6 for membrane obstacles.	22
3.14	Comparing maximal velocity from the first and second simulation. (a) Velocity at the center of the canal at 0.6 for radial obstacles. (b) Velocity at the center of the canal at 0.6 for membrane obstacles.	23

1 Introduction

1.1 Motivation

By enabling precise manipulation of fluids at small scales, microfluidics has led to major progress in fields such as molecular biology, synthetic chemistry, diagnostics, and tissue engineering. Specifically, the idea of controlling the flow parameters by careful design of the microfluidic channel has recently gained a lot of popularity. The advantage of this is that no active intervention is required to tune the flow, the intrinsic geometry of the channel results in the intended flow conditions. For example, Chan and Kanso use special geometries to create flow conditions for trapping and manipulating particles [2]. These microfluidic devices usually involve a large parametric space and exactly choosing the right parameters serves as the bottleneck in the design phase. The futility of achieving this by actually building the channel and then measuring the flow parameters renders simulations an indispensable part of the design phase.

1.2 Identification of the Scientific Question and needed Investigation

The motivation of this work is to investigate the change in velocity, pressure, and mass flow rate for different shapes of a canal as shown in Fig. 1.1. We create a rigid rectangular canal with an obstacle in the center top wall.

We model a no-slip boundary condition at the top and bottom walls of the channel by applying a dirichlet boundary condition to the velocity field at these locations. An inlet and outlet pressure are forced via dirichlet boundary conditions applied on the pressure field such that the pressure difference between inlet and outlet is ΔP_{DS} as shown in Fig. 1.1.

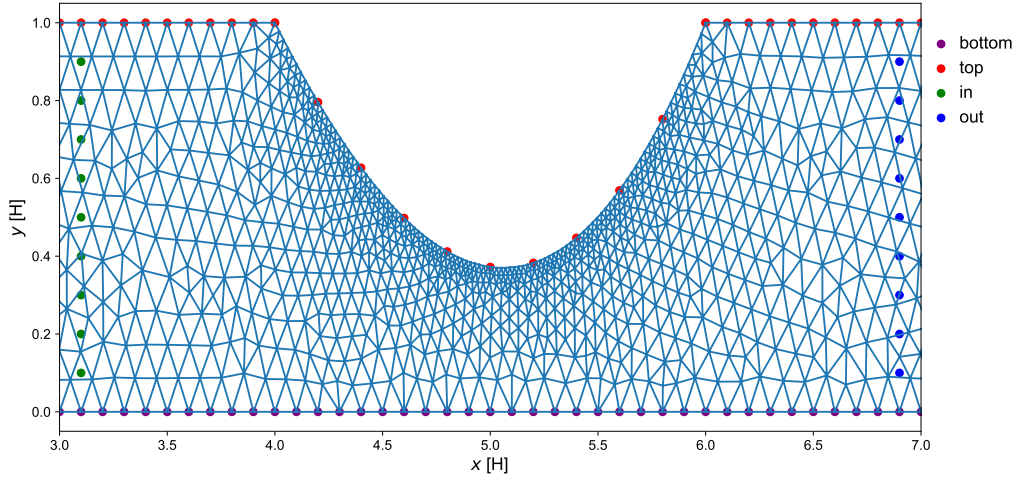


Figure 1.1: Part of a generated mesh of a canal with restricted flow showing top and bottom boundary conditions and indicating directions of in- and outflow.

The domain spanned by the canal is discretized on a mesh grid and solved in a 2-Dimensional simulation using the Finite element method (FEM) software DOLFINx. The simulation is controlled by the pressure ΔP_{SD} and shows the development of the Poiseuille flow, where we observe change of velocity and mass flow rate. We validate the results of the canal without an obstacle by comparing the developed Poiseuille flow to an analytical solution.

Using this approach, we conduct two parametric studies with different obstacle shapes. The first simulation was done using a semi circular shape with variable length depending on the radius. In the second simulation, we used a Ordinary Differential Equation (ODE) to create the shape of a membrane. The membrane was changed iteratively, controlled by ΔP_{SD} and the pressure P_0 at the membrane beginning and P_L at the end. Finally, we discuss the changes observed after incrementally adding the obstacles.

2 Theory

2.1 Governing Equations System

For the simulation of incompressible fluids in a canal with an obstacle we only need the governing equations of Navier Stokes Equation (N-S) regarding the conservation of mass and the conservation of momentum equations.

For our calculation we use the incompressible N-S

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \nabla) \cdot \vec{u} \right) = \nabla \cdot \sigma(\vec{u}, p) + \vec{f} \quad (2.1)$$

$$\nabla \cdot \vec{u} = 0 \quad (2.2)$$

We assume constant density ρ and the outer force f is assumed as 0. Further μ denotes the dynamic viscosity, p the pressure in the equations. For simplicity, we refrain from using a vector notation in further equations.

2.2 Equations

When modeling physical systems using mathematical formulas, differential equations are used. This is because we are primarily interested in understanding how these systems change over time, and differential equations provide models for capturing these dynamic changes. Although we can describe these systems in a model, the possibility of solving the model further depends on the characteristics of the Differential Equation.

2.2.1 General

We further introduce the necessary characteristics.

The **order** of the Ordinary Differential Equation (ODE) or Partial Differential Equation (PDE) describes the highest order of the dependent variable. In the case of N-S we are solving a second-order equation due to $\mu \nabla^2 u$ in the momentum equation.

Nonlinear PDEs are equations in which the dependent variable is dependent on itself or on a different dependent variable. In N-S the non-linearity is also part of the momentum equation as in $(u \nabla) u$. But as the coefficients of the highest order derivatives are

functions of only the dependent variable, the N-S are quasilinear.

The velocity field is described in two equations - the mass continuity and the momentum - thus **coupling** should be considered to solve N-S.

An analytical solution for the N-S is nonexistent and possible solutions heavily depend on boundary conditions and assumptions, such as incompressibility of the fluid, steady state, laminar flow, or simplified geometry.

2.2.2 Assumptions

We use the assumption of linear elasticity of a Newtonian fluid to write σ from eq. (2.1) as

$$\sigma(u, p) = \mu (\nabla u + (\nabla u)^T) - pI \quad (2.3)$$

We take the divergence of the stress σ

$$\nabla \cdot \sigma(u, p) = \nabla \cdot (\mu (\nabla u + (\nabla u)^T) - pI)$$

where I denotes the identity tensor and use eq. (2.2) to obtain.

$$\sigma = -\nabla p + \mu \nabla^2 u \quad (2.4)$$

The final expression for the N-S becomes

$$\rho \left(\frac{\partial u}{\partial t} + (u \nabla) \cdot u \right) = -\nabla p + \mu \nabla^2 u + f \quad (2.5)$$

$$\nabla \cdot u = 0 \quad (2.6)$$

Here eq. (2.5) represents conservation of momentum, while eq. (2.6) represents conservation of mass (continuity equation). These equations form the governing system for our incompressible fluid simulation.

2.2.3 Analytical solutions

We take the analytical expressions for pressure

$$p_x = \Delta P_{SD}(L - x) \quad (2.7)$$

and velocity

$$u_x = \frac{1}{2} \Delta P_{SD} y(1 - y) \quad (2.8)$$

from [1] to validate the correctness of our simulations without obstacles in the channel. ΔP_{SD} is the pressure gradient in the canal and the parameter, that is changed to obtain different profiles in our simulations.

2.2.4 Introduction of Dimensionless variables

To simplify the problem and make numerical solving possible in a numerical simulation, we introduce a non-dimensionalisation scheme as described in [1]. As the physics of channel flow are two-dimensional, the z-axis is omitted for the calculation leading to the assumption $\frac{\partial u_z}{\partial t} = 0$.

Further we use $\bar{x} = x/H, \bar{y} = y/H, \bar{z} = z/H, \bar{u} = u/U, \bar{p} = Hp/\mu U$ and $\bar{t} = (H^2t)/(\mu/\rho)$ as described in [1]. Therefore our equation eq. (2.5) is now denoted as

$$\begin{aligned} \frac{\partial u}{\partial t} + (u\nabla) \cdot u &= -\nabla p + \frac{1}{\mathbf{Re}} \nabla^2 u \\ \nabla \cdot u &= 0 \end{aligned} \quad (2.9)$$

where we omit the bars and vector notion and refer to the equation as dimensionless.

We assume μ, ρ , and \mathbf{Re} as unity. \mathbf{Re} is derived from the convective term and denoted to $\mathbf{Re} = \frac{\rho U H}{\mu}$.

2.2.5 Incremental Pressure Correction Scheme

The Incremental Pressure Correction Scheme (IPCS) is used in [1] to create the weak forms used in the simulation. The method is further described in [3]. It consists of three steps to calculate the incompressible N-S equation as shown in eq. (2.9). The most important achievement of this method is, the equations are split into two 1-Dimensional equations correcting the pressure and calculating the velocity, as we have seen in section 2.2.3 and are supplied with analytical solutions.

The first step consists of an integration, using an Euler scheme thereby splitting the equation in the current and next step denoted as u^n and u^{n+1} .

The equations are

$$\frac{1}{\Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n) - \nabla p^{n+1} = -\mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{1}{Re} \nabla^2 \mathbf{u}^n \quad (2.10)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (2.11)$$

where Δt denotes a time step.

When using Euler scheme, the conservation of momentum is not necessarily satisfied, thus an extra step needs to be taken by splitting p^{n+1} into

$$p^{n+1} = p^n + p^* \quad (2.12)$$

and an auxiliary velocity field $\hat{\mathbf{u}}$ and inserting into eq. (2.10) reads

$$\frac{1}{\Delta t} (\mathbf{u}^{n+1} - \hat{\mathbf{u}} + \hat{\mathbf{u}} - \mathbf{u}^n) - \nabla(p^n + p^*) = -\mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{1}{Re} \nabla^2 \mathbf{u}^n \quad (2.13)$$

As the next step and the first of the split equations, $\hat{\mathbf{u}}$ can be calculated from

$$\frac{1}{\Delta t} (\hat{\mathbf{u}} - \mathbf{u}^n) = -\nabla(p^n) - \mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{1}{Re} \nabla^2 \mathbf{u}^n \quad (2.14)$$

where

$$\frac{1}{\Delta t} (\mathbf{u}^{n+1} - \hat{\mathbf{u}}) = -\nabla p^* \quad (2.15)$$

is assumed. By taking the divergence of eq. (2.15) and requiring eq. (2.11), we can use the continuity constraint leaving us with following term for pressure

$$\nabla^2 p^* = \frac{1}{\Delta t} \nabla \hat{\mathbf{u}} \quad (2.16)$$

At last for the third step we need to calculate p^{n+1} using

$$\begin{aligned} \frac{1}{\Delta t} (\hat{\mathbf{u}} - \mathbf{u}^n) &= -\mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{1}{Re} \nabla^2 \mathbf{u}^n \\ \frac{1}{\Delta t} (\mathbf{u}^{n+1} - \hat{\mathbf{u}}) &= -\nabla p^{n+1} \end{aligned} \quad (2.17)$$

which is transformed to

$$\frac{1}{\Delta t} \nabla \hat{\mathbf{u}} = \nabla^2 p^{n+1} \quad (2.18)$$

using divergence to use p^{n+1} in eq. (2.17) to obtain the velocity for our next step.

2.2.6 Deflection of a membrane in a channel

To simulate the deflection of a membrane, we create the following model shown in Fig. 2.1. Depicted is a rectangular channel in black, the pressure in red with markings for inflow and outflow and the membrane in blue colors. 0 and L are the beginning and endpoints of the membrane. P_0 and P_L respectively stand for the pressure at the beginning and the end of the membrane.

From the model we can derive a simple ODE calculating the Pressure on the Membrane with

$$P(x) = P_0 - P_G + \frac{P_L - P_0}{L} x \quad (2.19)$$

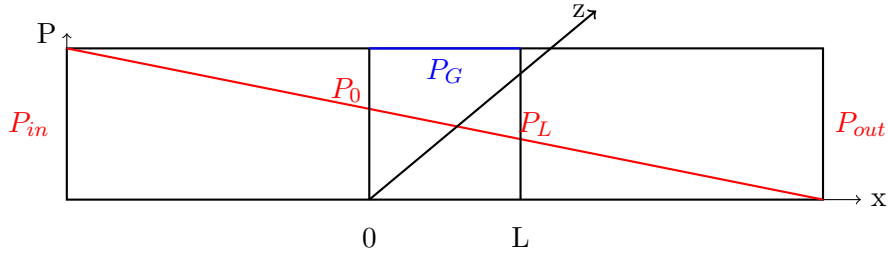


Figure 2.1: Rectangular Channel

The equilibrium form of eq. (2.19) can be written as a balance equation for small membrane deflections

$$T\nabla^2\zeta(x, z) + P(x, z) = 0 \quad (2.20)$$

where $\zeta(x)$ is the membrane deflection and T is the tension. As we simulate in 2-Dimensional space and therefore perpendicular to the plane, the z component can be neglected due to an infinite extension in the z direction resulting in

$$T\nabla^2\zeta(x) + P(x) = 0 \quad (2.21)$$

The equation can be solved with integration and the initial conditions 0 and L

$$\begin{aligned} \zeta(x) &= -\frac{1}{T} \iint P(x) dx \\ &= \frac{1}{T} \iint [P_G - P_0 + \frac{P_0 - P_L}{L}x] dx \\ &= \frac{1}{T} [\frac{1}{2}(P_G - P_0)x^2 + \frac{P_G - P_L}{6L}x^3 + c_1x + c_2] \end{aligned}$$

with

$$\begin{aligned} c_2 &= 0 \\ c_1 &= -\frac{1}{2}(P_G - P_0)L - \frac{1}{6}(P_0 - P_L) \\ &= -\frac{1}{6}(3P_G - 2P_0 - P_L)\frac{L}{T} \end{aligned}$$

resulting in

$$\zeta(x) = \frac{1}{T} [\frac{1}{2}(P_G - P_0)x^2 + \frac{P_G - P_L}{6L}x^3 - \frac{1}{6}(3P_G - 2P_0 - P_L)Lx] \quad (2.22)$$

2.3 Simulation

2.3.1 Stability - Courant-Friedrichs-Lewy condition(CFL)

The Courant-Friedrichs-Lewy condition (cfl) is defined as

$$c = \frac{u\Delta t}{\Delta x} \quad (2.23)$$

and provides an idea of discretizing the problem in time and space for the calculation. A courant number $c \leq 1$ is said in [4] to be stable in fluidic simulations.

2.3.2 Units

The units in the simulation as described in Introduction of Dimensionless variables from [1] are dimensionless. This approach offers several advantages, as the N-S have fewer terms to solve thus simplifying the computational effort. The simulation is generalized, thereby we obtain relations of parameter ranges for the optimal results and thus can apply them on different materials and conditions.

2.3.3 Mesh

The mesh for the simulations is created using the software package gmsh. As described by Geuzaine and Remacle in [5] gmsh is a open-source 3D finite element mesh generator with multiple integrated CAD engines, which in turn can be seamlessly imported into DOLFINx for numerical calculation. Gmsh further offers fine-grained control options over the mesh density overall and at specific areas, resulting in more control over calculation intensity for important points. This enabled our simulations to have dense regions for the radial obstacle and the membrane to capture precise results for pressure, velocity and mass flow rate.

2.3.4 Boundary conditions

2.3.4.1 Inflow and outflow

To drive the flow, we establish a pressure gradient along the channel by specifying different pressures from inlet to outlet. At the inlet ($x = 0$), we set a higher pressure. The given eq. is used in [1] and is modified to enlongate the canal with pressures

$$p = L\Delta P_{SD} \text{ at } x = 0$$

and

$$p = 0 \text{ at } x = L$$

In our simulations we extended the channel length to $L = 10$

This pressure difference creates a driving force that initiates and maintains the flow through the channel. The resulting flow is expected to develop a characteristic parabolic velocity profile, with the maximum velocity occurring at the centerline of the channel ($y = 0.5$).

2.3.4.2 Dirichlet Boundary Conditions

For the velocity field, we use no-slip Dirichlet boundary conditions as described in [1] on the channel walls. Mathematically, this is expressed as $\mathbf{u} = (0, 0)$ at $y = 0$ and $y = 1$ for all $x \in [0, 10]$. This condition ensures that the fluid cannot move along the channel walls and enforces a velocity of 0 at the walls.

2.3.5 DOLFINx

DOLFINx is a Python package combining many software to provide a way of solving PDE as described in [1]. It provides the tools to create meshes, function spaces, variational formulations and many solvers implemented mostly in C++. We use it as described in [1] to solve the incompressible N-S and will introduce the key points next.

2.3.5.1 Code implementation of Finite element method (FEM)

The IPCS delivers the equation we want to solve with FEM in the dolfinx-computation. Hence, the first step is

```
F1 = rho * dot((u - u_n) / k, v) * dx
F1 += rho * dot(dot(u_n, nabla_grad(u_n)), v) * dx
F1 += inner(sigma(U, p_n), epsilon(v)) * dx
F1 += dot(p_n * n, v) * ds - dot(mu * nabla_grad(U) * n, v) * ds
F1 -= dot(f, v) * dx
a1 = form(lhs(F1))
L1 = form(rhs(F1))
```

where k is the time step $\frac{1}{\Delta t}$, v represents the test function used in the FEM for the velocity and q respectively the test function for the pressure. This is the code representation of eq. (2.14). *sigma* is the stress tensor and *epsilon* the strain-rate tensor which are interchangeable non-dimensionalisations as described in section 2.2.2. We point out, that the notion of eq. (2.1) is used as the impact of addition and multiplication in numerical solutions on the calculation time is neglectable.

Consequently

```
u_ = Function(V)
a2 = form(dot(nabla_grad(p), nabla_grad(q)) * dx)
L2 = form(dot(nabla_grad(p_n), nabla_grad(q)) * dx - (rho / k) * div(u_) * q * dx)
```

represents eq. (2.16) with $u = u^{n+1}$ and $p = p^{n+1}$ and

```
p_ = Function(Q)
a3 = form(rho * dot(u, v) * dx)
L3 = form(rho * dot(u_, v) * dx - k * dot(nabla_grad(p_ - p_n), v) * dx)
```

the last step as in eq. (2.18). The term *Function* forms matrices for the calculated values, hence at every time step we have a solution at every point of the defined domain.

As the code is largely described in [1] we abstain of further details.

2.3.5.2 Calculating massflowrate

We developed two methods to calculate the mass flow rate from the results. The first method relies on the dolfinx implementation of integrating

```
# Function calc_mfl(...):
[...]
u_sub = u_n.sub(0)
# Define measures and spatial coordinates
dx = Measure("dx", domain=mesh)
x = SpatialCoordinate(mesh)
tol = 5e-2
mfl, mass_flow, p_loc, pressure_avg = np.array([]), None, None, np.array([])
for i in np.array([0.5, x_max/2, x_max-.5]):
    slice_condition = conditional(ge(x[0], i-tol), 1.0, 0.0) *
        conditional(le(x[0], i+tol), 1.0, 0.0)
    # Calculate mass flow rate at the current slice
    mass_flow_local = assemble_scalar(form(u_sub * slice_condition * dx))
    mass_flow = mesh.comm.allreduce(mass_flow_local, op=MPI.SUM)
    mfl = np.append(mfl, mass_flow)}
[...]
```

where a slice of the domain at the beginning, the center - also the location of the obstacle - and the end were calculated with

$$Q = \int_{\Omega} \rho u_x S(x), d\Omega \quad (2.24)$$

where u_x represents the velocity at x , $S(x)$ the slice of the domain at x and $d\Omega$ the domain, respectively the mesh we want to use for calculation.

The second method leverages *np.trapz* to calculate the integral of the velocity. We used theses two method concurrently, both inside the main loop of the calculation as

```
flux = np.array([np.trapz(y=y1[:,0],x=y_grid),
                 np.trapz(y=y2[:,0],x=y_grid2),
                 np.trapz(y=y3[:,0],x=y_grid3)])
```

where we previously obtained the coordinates of $y \in [y1, y2, y3]$ and $x \in [y_grid, y_grid2, y_grid3]$ for the different heihgts and locations.

2.4 Dtool

For saving, managing and sorting the data obtained in simulations, *dtool* [6] was used. *dtool* is a Python package with CLI-API to manage data programmatically and was used in this work to save simulation data to create figures.

Each dataset of one simulation run saves information like pressure \bar{p} and \bar{u} from a dedicated time step as a slice of the domain, the mass flow rates at different points of the domain, as well as the height, length, time step and discretization size of the mesh.

An example of the data and structure is provided in chapter 4.

3 Results and Discussion

We conducted two different simulations with varying pressure difference between in- and outlet and different forms of obstacles in the canal. The simulations are run as a parametric study for pressure, obstacle size, velocity and mass flux. The first simulation is run with a radial obstacle in the center of the canal and the second with a functional description of a membrane deformation based on the pressure drop throughout the canal as shown in (2.22). The code can be found at github¹.

3.1 Results

3.1.1 Finding suitable parameter dimensions

First we extended the length of the canal in contrast to [1] to 10 [H]. At shorter canal lengths we were experiencing backflow. This led to a negative velocity at the top of the canal beginning at the obstacle and going back to the inflow, thereby rendering the result of the Poiseuille flow unusable. With this change, we had to increase the pressure P_{SD} using $L \cdot P_{SD}$. We measured the mass flow rate and pressure for different P_{SD} and observed linear increases of mass flow rate with rising P_{SD} . A saturation of the mass flow rate could not be simulated.

We decided on pressure values between 120 and 2000 ΔP_{SD} for the first simulation and obstacles up to 60 % of the height of the canal to ensure stability of the simulation.

The second simulation was more complex to handle, as the membrane deformation calculated with (2.22) does not have a preset of obstacle height in percentage but instead is dependant on P_0, P_g, P_L, T and L from the previous simulation run as shown in Fig. 2.1. We set T to 1000 to maximize the mass flow rate in our simulations, resulting in a maximal membrane height of up to 34 %.

The mesh size was chosen between 0.02 and 0.03 and Δt was $\frac{1}{2000}$ in the first and $\frac{0.8}{1000}$ in the second simulation leading to courant numbers c from (2.23) between 0.28 and 0.33.

Trying to find the optimal pressure range, we noticed crashes of the simulations in high values of ΔP_{SD} . The same was the case for radial obstacles ranging between 60 to 75 % of the height H .

¹<https://github.com/SickSmile1/bsc-ilia-fedotov>

3.1.2 Parametric study with half circle as obstacle

As the first simulation, we used a half-circle obstacle at the center of the canal. The obstacle radius was varied from 0 to 0.6. For each obstacle radius we run 15 simulations with different pressure from 120 to 2000 ΔP_{SD} . The mass flow rate and pressure were measured at the beginning, center and end of the canal. We observed the velocity in our runs and noticed, that the profile is developed and does not significantly change after 0.4 simulation time. According to this knowledge, we observe plots at 0.1 and 0.6 for our results.

In Fig. 3.1 we can observe the change of the mass flow rate from the beginning of the simulation runs in Fig. 3.1a and the end of the simulation in Fig. 3.1b. The axis show the parameters of the parametric study, the obstacle radius r [H] and the pressure P_{SD} . The contour emphasizes the rising mass flow rate in steps of 1 in Fig. 3.1a and 2 in Fig. 3.1b.

A clear trend of rising mass flow rate with small radii and without obstacle in both Fig. 3.1a and Fig. 3.1b is present. The mass flow rate rises from ≈ 10.7 to ≈ 16.7 comparing Fig. 3.1a and Fig. 3.1b at the simulation without an obstacle. For a radius of 0.6 the change in mass flow rate is barely noticeable in the range of ≈ 6.34 to 7.41. At a pressure P_{SD} of 2000 we observe a mostly linear rise of mass flow rate from an obstacle of 0.6 to the maximal value without an obstacle. There is a slight bend in the values, observable in the contour of the plots visible between radii of 0.06 and 0.09 from a pressure of 700 to 2000 P_{SD} .

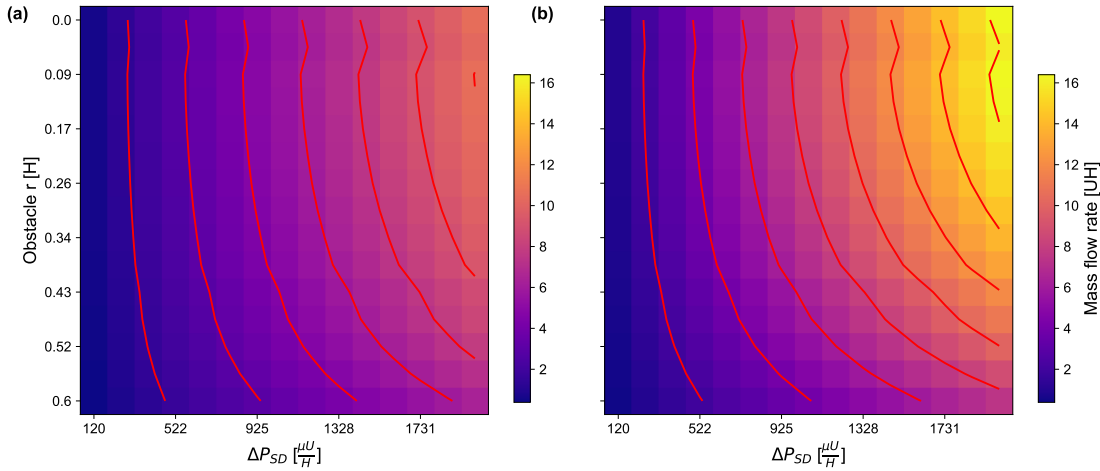


Figure 3.1: Mass flow rate for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.

We can see in Fig. 3.2 that the rise of mass flow rate is linear in respect to the change in pressure. Higher obstacle radii begin at lower values for mass flow rate and have

a smaller maximum, compared to smaller obstacle radii. The canal without obstacles shows the greatest growth rate as well as greatest minima and maxima throughout our simulation range.

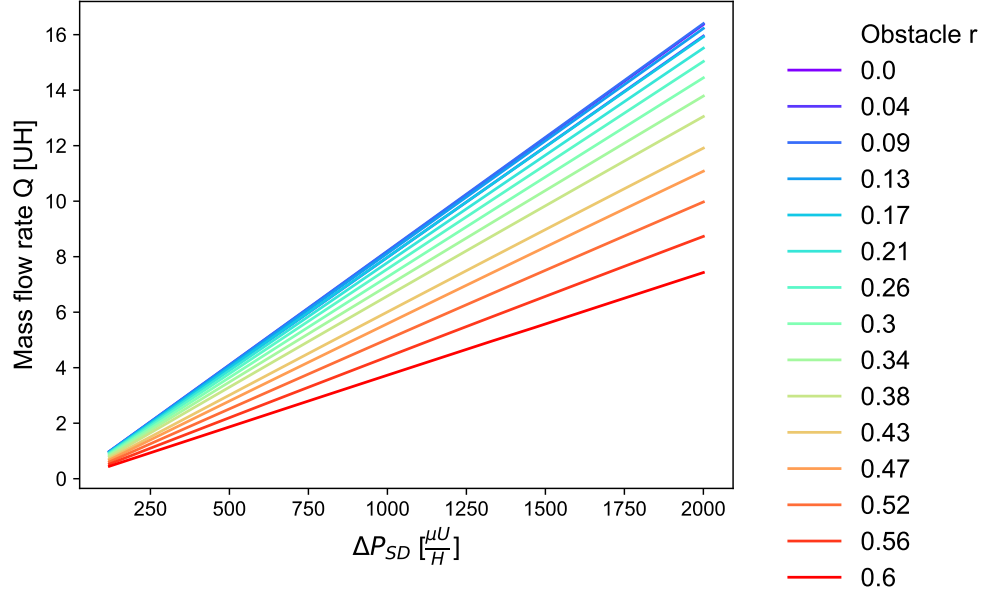


Figure 3.2: Mass flow rate in the center of canal over pressure difference at 0.6.

In Fig. 3.3 we can observe the change of pressure at 0.1 in Fig. 3.3a and at 0.6 Fig. 3.3b. The contour shows emphasizes the growth of pressure in steps of 150.

We can observe a trend of rising pressure with small radii and without obstacle in both, the beginning and end of the simulation. The shown values without obstacle at $r = 0$ correspond to roughly $\frac{1}{2}\Delta P$ and are linear. The pressure in simulations with ΔP over 900 increase slower for radii over 0.34 [H] and range from 60 to 800. The pressure along the y-axis showing the obstacle radius [H] is ≈ 60 for all obstacle sizes.

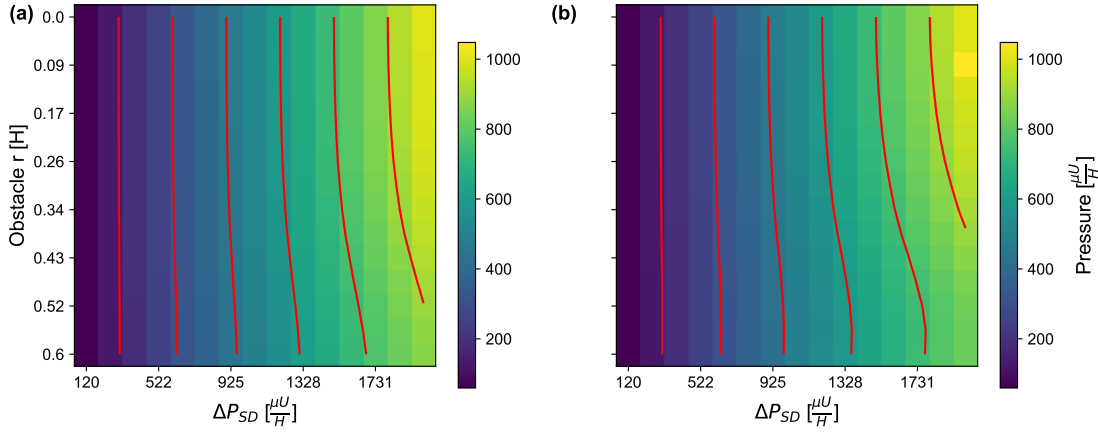


Figure 3.3: Pressure for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.

The velocities can be seen in Fig. 3.4. In the beginning of the simulation we observe a linear growth with bigger radii of obstacles and rising pressure. Without obstacle, we observe a raise from 1 to 15 U with pressure P_{SD} of 120 and respectively from 15 and 27 U for pressure of P_{SD} of 2000. At later timesteps the curves shift, showing maximal velocity for the ΔP of 2000 and obstacle radii between 0.3 and 0.48. In Fig. 3.4b radii between 0.34 and 0.43 all reach a velocity of 30 U.

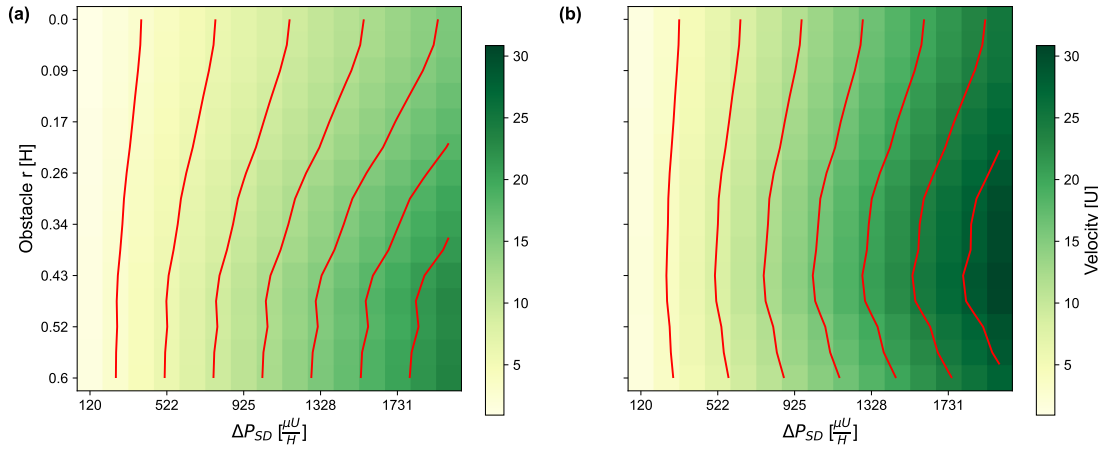


Figure 3.4: Velocity for parametric run of different radii and pressures in the center of the canal at 0.1 (a) and 0.6 (b) simulation time.

3.1.3 Parametric study with iterative membrane as obstacle

In our last simulation, we iteratively changed the membrane by increasing the pressure P_G on the membrane for each individual run. Thereby, we obtained data for a dedicated

P_{DS} at different pressures P_G on the membrane and run our simulations with the membrane shapes shown in Fig. 3.5a with the resulting velocity profiles shown in Fig. 3.5b. The chosen parameters allow for a parametric study for P_G ranging from 600 to 1000 as we had to fixate the T from (2.22) to 1000 to ensure the membrane is stiff enough to reach the obtained maximal velocities of the radial parametric run of 12 U.

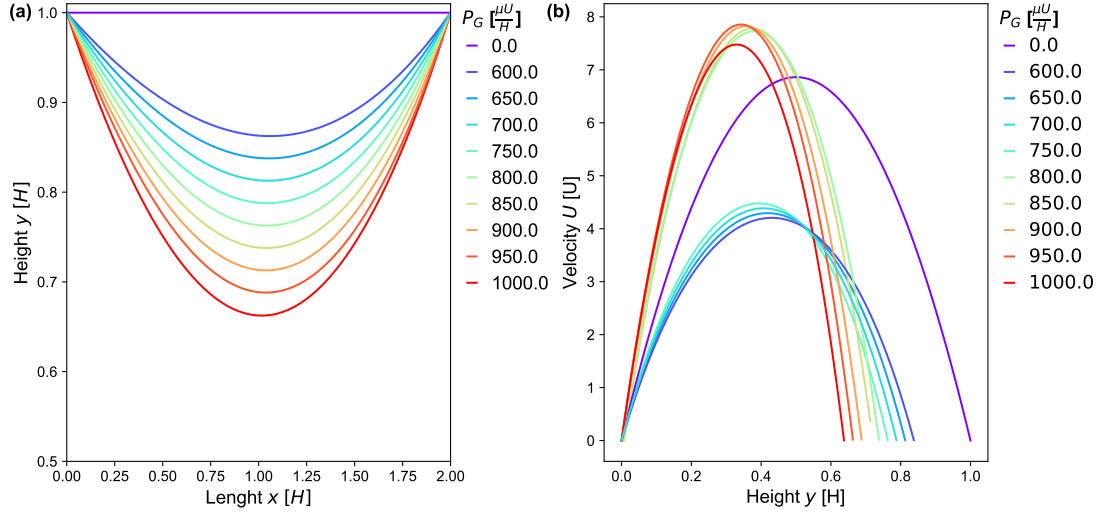


Figure 3.5: Membrane (a) shapes and (b) velocities for parametric membrane simulation.

In Fig. 3.6a and Fig. 3.6b we can observe the results of the parametric runs for mass flow rate and velocity for different P_{DS} on the x-axis and P_G on the y-axis. The contour emphasizes the change of mass flow rate in steps of ≈ 0.25 . The mass flow rate is highest for a canal without membrane. We observe a decrease of mass flow rate with greater values of P_G and thus greater obstacles.

The velocity under the obstacle behaves opposite to the mass flow rate - it rises with bigger obstacle. The lowest velocity rises from the smallest membrane and P_G and towards greater values of P_G and greater membranes sizes. The velocity without a membrane is on the upper end of the measured values and rises linearly with approximately 6 U.

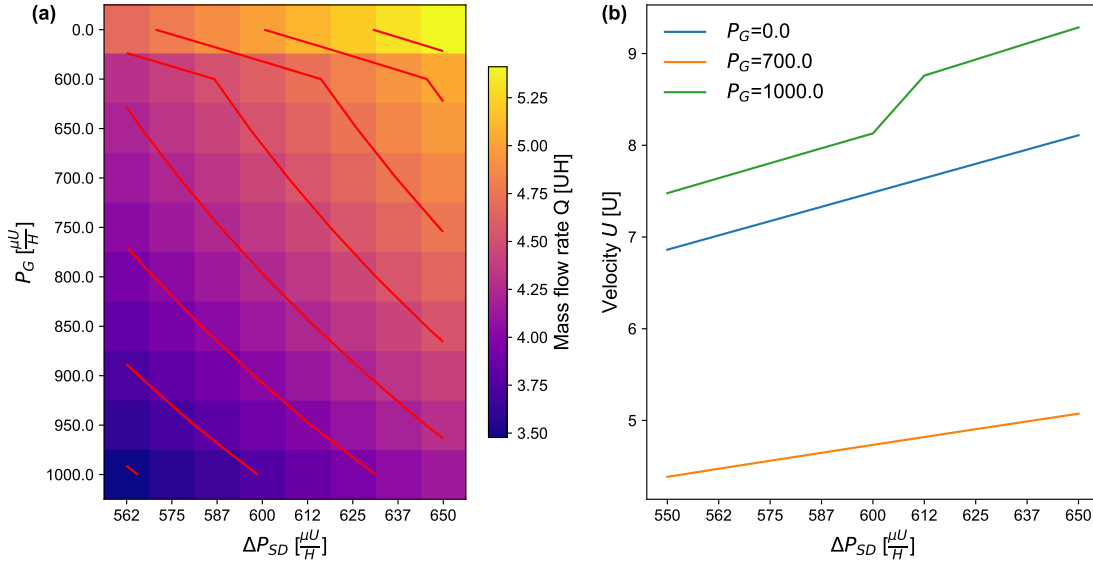


Figure 3.6: Change in mass flow rate and velocity for different P_G and membranes. (a) The change in mass flow rate. (b) The change of velocity.

In Fig. 3.7 we can observe the pressure drop over the membrane. The indicated numbers were also used for the next consecutive run with the change of P_G using the first value at 0 as P_0 and the last at 2 as P_L . We can observe a slightly sinusoidal shape for greater values of P_G , where the beginning of the curves lacks the rise.

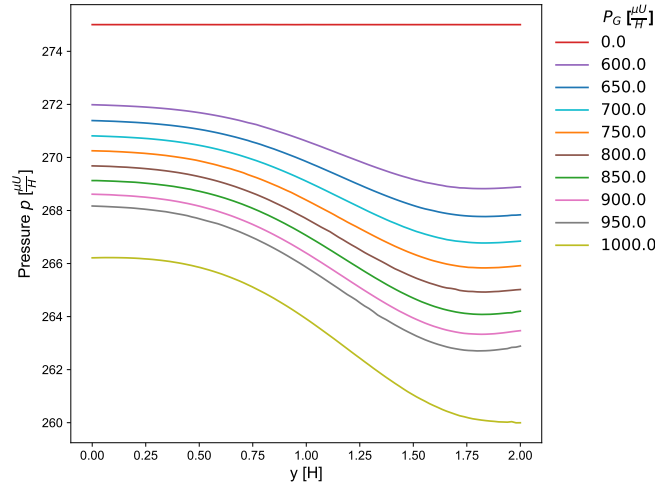


Figure 3.7: Pressure over membrane length of 2 at 0.32 with P_{SD} of 550.

The pressure in the center of the canal is shown in Fig. 3.8 and rises linearly throughout changes to P_G and P_{SD} . The pressure ranges from ≈ 260 to ≈ 330 in $\frac{HP}{\mu U}$.

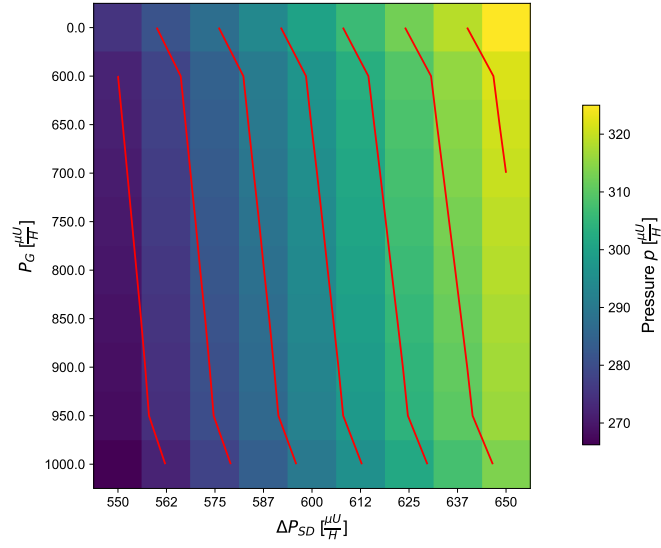


Figure 3.8: Pressure in the center of the canal at 0.32.

In Fig. 3.9 we can observe the change of mass flow rate over the different pressure P_{DS} . The mass flow rates rise linearly and show the greatest values for canals without membrane and lower values for greater P_G and membrane obstacles.

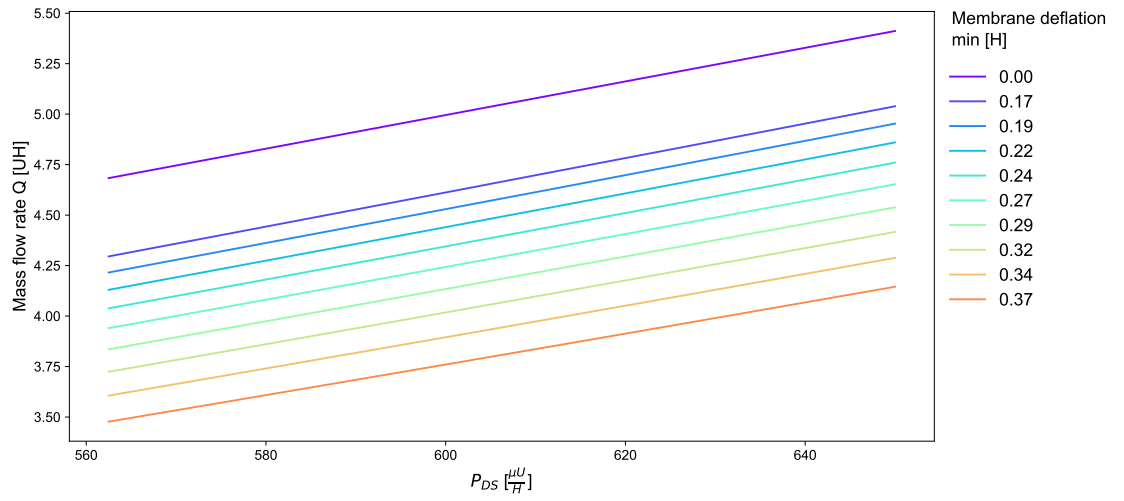


Figure 3.9: Mass flow rate in the center of canal over pressure difference at 0.32.

3.2 Discussion

In the following section we compare the obtained mass flow rate, pressure and velocity from the two simulation runs.

For both simulation, the comparison is made in the coinciding range between the P_{DS} of 550 and 650 and focuses on obstacle radii ranging from 0 to 0.4 as the obtained results of the first parametric simulation run indicate symmetry or clear trends in this range.

First we compare the analytical values and the simulated values in Fig. 3.10. The mean distance between the analytical and simulated values is 2.08, the euclidean norm distance is 0.626 and the deviation percentage is 4.26 %. The analytical mass flow rate was calculated using (2.8) and `np.trapz` practically the same way, as it was used in the main loop of the simulation.

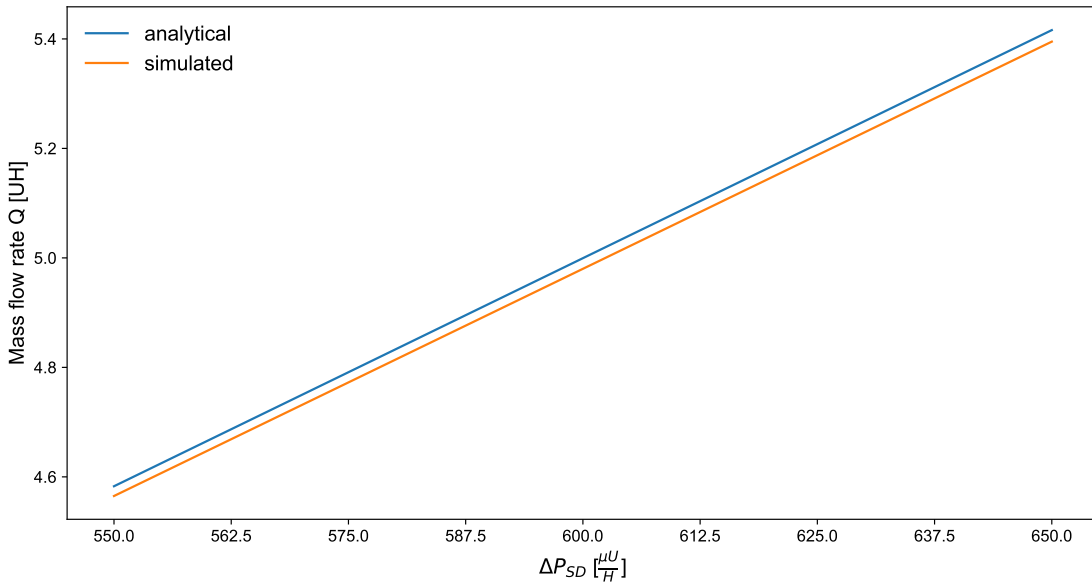


Figure 3.10: Analytical mass flow rate for different P_{SD} compared to simulated.

The comparison of the analytical and simulated velocity also is very close to each other. The mean distance is around 0.014, the euclidean norm distance 0.44 the deviation percentage is 1.95 %.

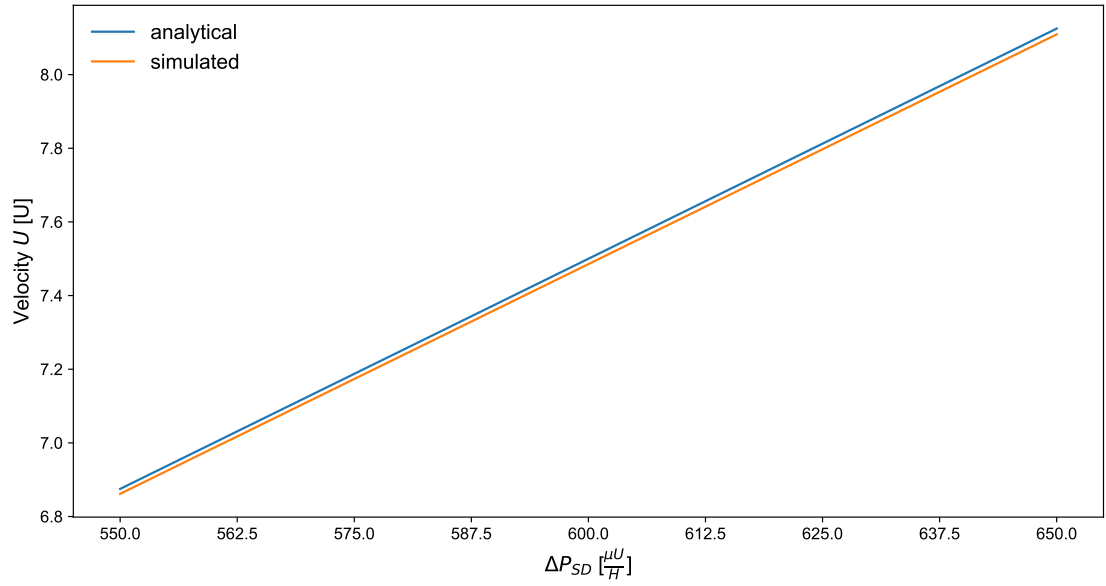


Figure 3.11: Analytical velocity for different P_{SD} compared to simulated.

3.2.1 Mass flow rate

For both simulations the mass flow rate behaves as expected. We observe linear increase where the mass flow rate increases by 20 % with smaller obstacle radii and ≈ 25 % for bigger radii. The increase of mass flow rate for different membranes remains at ≈ 20 % throughout our simulations.

The smallest membrane run with an absolute membrane height of 0.14 can be compared to a membrane obstacle size of 0.3 from Fig. 3.12a. For depicted range of P_{SD} the mass flow rate is similar. This discrepancy can be explained by the fact, that the membrane obstacle has a fixed length of 2 compared to a length of the radial obstacle of $2 \cdot r = 2 \cdot 0.3 = 0.6$. This leads to comparable mass flow rate values between 4 and 4.5. This is also the reason for shrinking rate of mass flow for radial obstacles with greater radii.

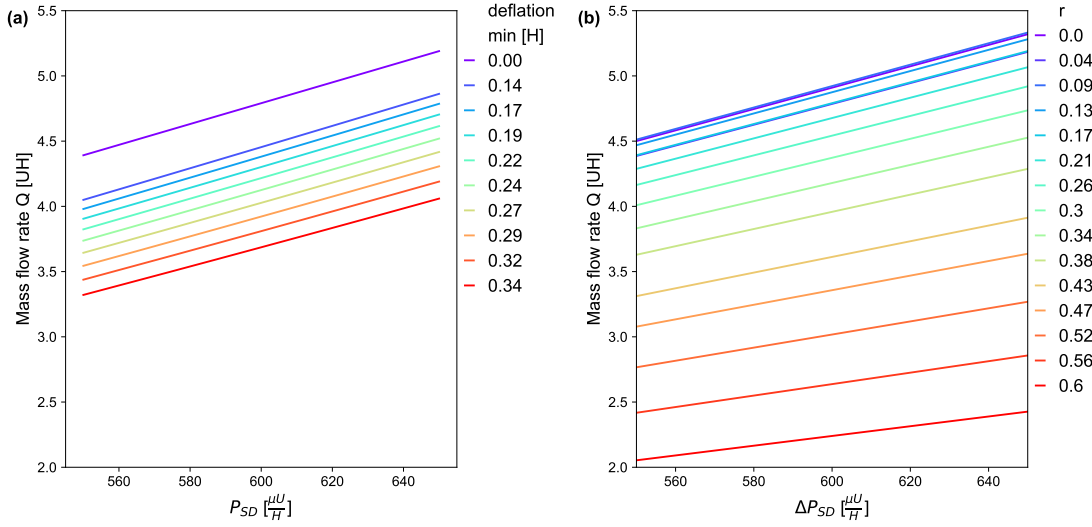


Figure 3.12: Change in mass flow rate for different P_G and radii compared to different mass flow rates of the membrane simulation. **(a)** Change in mass flow rate for different P_G and radii compared to different mass flow rates of the membrane simulation. **(b)** The change of mass flow rate for different P_G and membranes.

3.2.2 Pressure

Comparing the pressure at the center of the canal, we observe they coincide for Fig. 3.13a and Fig. 3.13b. Further, we expected the pressure to have the value of $\frac{1}{2}\Delta P_{DS}$ at the center of the canal. For the values without obstacle we observed values of 261.4 and 328.5 in our simulations coinciding with the $\frac{1}{2}\Delta P_{DS}$. With greater obstacle sizes, the measured pressure dropped, respectively to 249.3 and 309.5 at a radius of 0.6. Overall the results from the second simulation are observed as more linear with greater membrane obstacles in comparison to the first simulation.

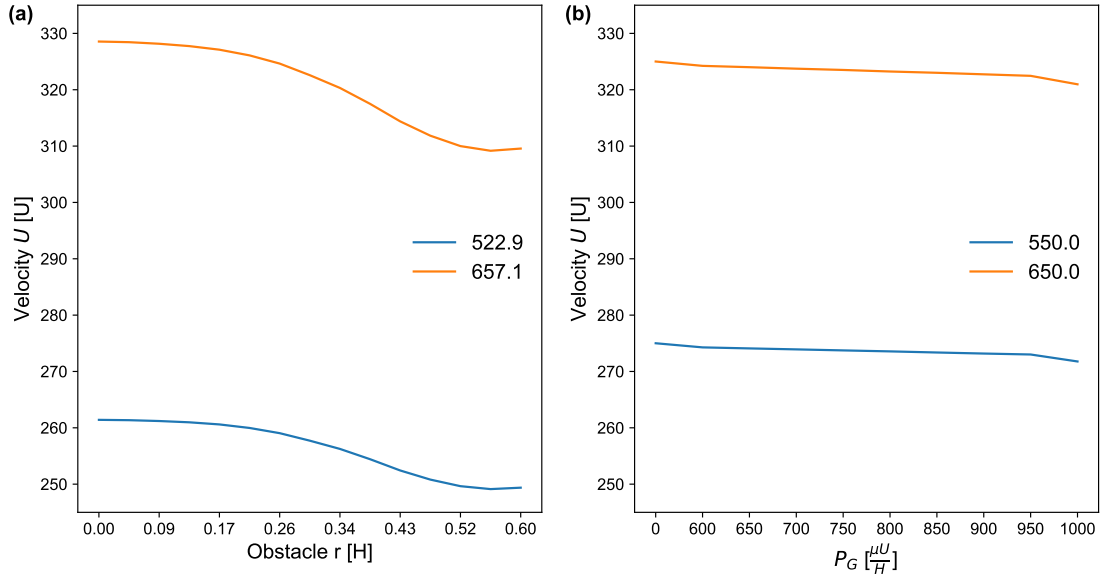


Figure 3.13: Comparing pressure from the second and third simulation. **(a)** Pressure at the center of the canal at 0.6 for radial obstacles. **(b)** Pressure at the center of the canal at 0.6 for membrane obstacles.

3.2.3 Velocity

The velocity shown in Fig. 3.14 behaves reverse to the mass flow rate. We observe a rise from no restriction to the biggest restriction with radial obstacle. The membrane obstacle shows a cosinusoidal shape. It is low at no obstacle, then falling and at last rising with membrane minima of 0.2 [H] and higher. The velocity increase observed in the second simulation doesn't quite reach the increase achieved in the first simulation. In the first simulation the velocity increases ≈ 24 % where in the second simulation we observe an increase of ≈ 21 %. We consider this deviation small enough to conduct the reason being the difference in P_{SD} between the first and second simulation from 522 to 550.

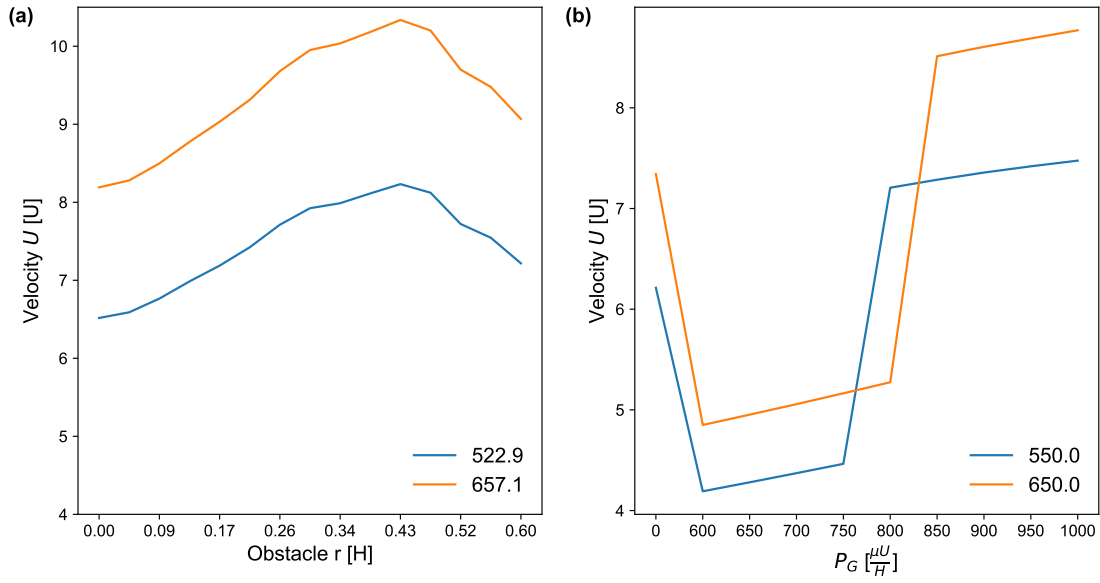


Figure 3.14: Comparing maximal velocity from the first and second simulation. **(a)** Velocity at the center of the canal at 0.6 for radial obstacles. **(b)** Velocity at the center of the canal at 0.6 for membrane obstacles.

4 Summary and Conclusion

The values for mass flow rate, pressure, and velocity have been investigated on rigid rectangle canals with an obstacle of circular shape and a membrane deforming through pressure adjusted by the previous simulation run. The results have slight variation depending on the kind of obstacle, but overall indicate, that the efficiency or amplification of mass flow rate and pressure would most probably be achieved if the flow is restricted by $\approx 35\%$.

For comparison with [7] the incompressible Navier Stokes Equation (N-S) is not suitable. Many calculations, most important the intrinsic gain, are not possible due to incompressibility constraining local changes in mass flow rate. Furthermore, non-dimensionalization and pressure-driven calculation extend the simulation time unnecessary. Thus, for further simulation and parametric study, a velocity-driven approach with a fully developed velocity profile is of significant advantage compared to the method proposed in our work.

Appendix

metadata.json:

```
[
  {
    "height": 1,
    "length": 10,
    "pressure_delta": 550.0,
    "simulation_time": 0.4,
    "steps": 1000,
    "obstacle_location_x": 5.0,
    "meshing_size/tol": 0.02,
    "p0": 358.5382466336509,
    "p1": 192.48386485669042,
    "pg": 550,
    "membrane_factor": 1000,
    "accuracy zetta": 500
  }
]
```

tree data/

data

```
|— flux_trapz_0.08
|   |— flux_trapz.txt
...
|— flux_trapz_0.32
|   |— flux_trapz.txt
|— massflowrate_0.08
|   |— massflowrate.txt
...
|— massflowrate_0.32
|   |— massflowrate.txt
|— mesh.msh
|— p_at_0_0.08
|   |— p_at_0.txt
...
|— p_at_0_0.32
|   |— p_at_0.txt
```

```

├── p_at_1_0.08
│   └── p_at_1.txt
...
├── p_at_1_0.32
│   └── p_at_1.txt
├── p_at_5_0.08
│   └── p_at_5.txt
...
├── p_at_5_0.32
│   └── p_at_5.txt
├── p_courve_0.08
│   └── p_courve.txt
...
├── p_courve_0.32
│   └── p_courve.txt
├── y_at_0_0.08
│   └── y_at_0.txt
...
├── y_at_0_0.32
│   └── y_at_0.txt
├── y_at_1_0.08
│   └── y_at_1.txt
...
├── y_at_1_0.32
│   └── y_at_1.txt
├── y_at_5_0.9_0.08
│   └── y_at_5_0.9.txt
...
├── y_at_5_0.9_0.32
│   └── y_at_5_0.9.txt
├── y_courve_0.08
│   └── y_courve.txt
...
└── y_courve_0.32
    └── y_courve.txt

```

41 directories, 41 files

Acknowledgements

The author wish to thank Dr. A. Greiner for suggesting the topic of this work; J. L. Hörmann for his programmatical and technical assistance throughout the project and I. Ghanem and J. L. Hörmann for proof-reading the manuscript.

Bibliography

- [1] Anders Logg Hans Petter Langtangen. *The FEniCS Tutorial I*. Springer Verlag, 2016, pp. 56–65. DOI: <https://doi.org/10.1007/978-3-319-52462-7>. URL: <https://link.springer.com/book/10.1007/978-3-319-52462-7>.
- [2] Fan Kiat Chan and Eva Kanso. “Flow and particle manipulation across flow regimes via shape modulation”. In: *Physics of Fluids* 35.8 (2023), p. 083109. DOI: 10.1122/8.0000600.
- [3] Katuhiko Goda. “A Multistep Technique with implicit Difference Schemes for Calculating Two- or Three-Dimensional Cavity Flows”. In: *Journal of Computational Physics* 30 (1979), pp. 76–95.
- [4] R. Courant, K. Friedrichs, and H. Lewy. “Über die partiellen Differenzengleichungen der mathematischen Physik”. German. In: *Mathematische Annalen* 100.1 (1928), pp. 32–74. DOI: 10.1007/BF01448839.
- [5] Christophe Geuzaine and Jean-François Remacle. “Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.
- [6] Tjelvar S. G. Olsson and Matthew Hartley. “Lightweight data management with dtool”. In: *PeerJ* 7 (Mar. 2019), e6562. DOI: 10.7717/peerj.6562.
- [7] Kaustav A. Gopinathan et al. “A microfluidic transistor for automatic control of liquids”. In: *Nature* 622.7979 (2023), pp. 103–109. DOI: 10.1038/s41586-023-06517-3.