rijksuniversiteit groningen
faculteit Wiskunde en Natuurwetenschappen

Flow through a pipe with a sudden expansion; a comparison between MUMPS and Ilupack

Master research Applied Mathematics

At the University of Nottingham-School of Mathematical Sciences

July 2013

Student: J.G.I.Heijnen

First supervisor: dr. ir. F.W. Wubs (RuG)

Second supervisor: dr. E.J.C. Hall (UoN)

Third supervisor: prof. K.A. Cliffe (UoN)

Flow through a pipe with a sudden expansion; a comparison between MUMPS and Ilupack

J.G.I. Heijnen

5th July 2013

Abstract

In this research the flow of an incompressible fluid through a 1:2 sudden expansion of a pipe was modelled. The first model consisted of a symmetry-based reduction to a 2D calculation, and the second model used a Fourier spectral method in the azimuthal direction. MUMPS, a direct solver, was compared to Ilupack, a Krylov subspace based iterative solver. Both solvers were implemented in the AptoFEM package. The MUMPS solver was faster in all calculations. For the 2D calculations, MUMPS was significantly faster, as was expected since the number of degrees of freedom in these calculations were reasonably low. For the 3D calculations, MUMPS solving itself is still faster than Ilupack, but since the linear FE solver time for MUMPS is much higher than for Ilupack, the calculation times for MUMPS and Ilupack 3D calculations were similar, and in some cases the total Ilupack calculation time was actually lower than that of MUMPS.

Contents

1 Introduction

The flow of a fluid through a cylindrical pipe with a sudden expansion is of importance and interest due to its similarity to many applications, for instance in health, where locally narrowed blood vessels are modelled, and in industry. At first, flow through a 3D pipe, due to symmetry reduced to a 2D channel, will be simulated, later on we will consider the 3D case.

In section 2 a short overview of the Newton method, Finite Element Methods and the Spectral methods is given. A short introduction to MUMPS and Ilupack, the solvers used, is given in section 3. This is followed by an overview of the results, including tables, in section 4. And finally, in section 5 a short conclusion and discussion can be found. Additional information can be found in the appendices.

1.1 General Navier-Stokes equations

Fluid flow can be described by the Navier-Stokes equations. The multi-dimensional Navier-Stokes equations, describing the motions of an incompressible Newtonian fluid, are given by:

$$\begin{aligned} \rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) - \mu\nabla^2\mathbf{u} - \mathbf{f} + \nabla p = \mathbf{0}, \\ \nabla\cdot\mathbf{u} = 0. \end{aligned} \tag{1}$$

Here ρ is the density of the fluid, $\mathbf{u}$ is the velocity vector, p is the pressure, t is the time variable, μ is the dynamic viscosity and $\mathbf{f}$ represent the body forces. The gradient of $\mathbf{u}$ is with respect to space variable $\mathbf{x}$. Body forces are forces per unit volume [17], an example of which is gravity. The first equation is derived by using the physical concept of conservation of momentum in a closed system and the second one, often called the continuity equation, originates from conservation of mass [17].

An incompressible fluid is a fluid for which the density is constant: the volume of the fluid is not influenced by the pressure.

The Reynolds number is a dimensionless number and it is a measure for the ratio of viscous and inertial forces (the force required to make the flow stop). A high Reynolds number (of the order of several thousand) indicates turbulent flow, a low Reynolds number indicates laminar flow. An expression for the Reynolds number, for the geometries of Figure 2 on page 7, and Figure 1 on page 5 is given by:

$$Re = \frac{\rho u_{\max} R_1}{\mu} = \frac{u_{\max} R_1}{\nu}. \tag{2}$$

Here ρ is the density of the fluid, $u_{\max}$ is the maximum velocity of the incoming flow of the fluid, R is the radius of the pipe, μ is the dynamic viscosity and ν is the kinematic viscosity. The dynamic viscosity is a measure of a fluids resistance to flow, and the kinematic resistance is the ratio of the dynamic viscosity to the density, the kinematic viscosity is expressed for a volume instead of for a mass. It should be noted that different authors might use different definitions of the Reynolds number (using for instance the diameter instead of the radius, or the average velocity instead of the maximum velocity), see also Appendix A.

1.2 Equations and geometry

1.2.1 Incompressible Navier-Stokes equations

The steady state Navier-Stokes equations in cylindrical coordinates for the incompressible flow of a fluid through a pipe with sudden expansion are given by equations (3) and (4) (see Figure 1). Steady state indicates the solution does not depend on time. In Appendix B the coordinate transform from Cartesian to cylindrical coordinates is shown. The domain is given by $\Omega \times \Theta = (\Omega_1 \cup \Omega_2) \times \Theta = ([0, R_1] \times [0, L_1] \cup [0, R_2] \times [L_1, L_2]) \times [0, 2\pi\rangle$, and the forcing function equal to 0.

$$\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t} + u_r\frac{\partial u_x}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_x}{\partial \theta} + u_x\frac{\partial u_x}{\partial x}\right) + \frac{\partial p}{\partial x} \\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_x}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} + \frac{\partial^2 u_x}{\partial x^2}\right) = 0, \\
\rho\left(\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} + u_x\frac{\partial u_r}{\partial x} - \frac{u_\theta^2}{r}\right) + \frac{\partial p}{\partial r} \\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} + \frac{\partial^2 u_r}{\partial x^2} - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right) = 0, \\
\rho\left(\frac{\partial u_\theta}{\partial t} + u_r\frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta} + u_x\frac{\partial u_\theta}{\partial x} + \frac{u_r u_\theta}{r}\right) + \frac{1}{r}\frac{\partial p}{\partial \theta} \\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\theta}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} + \frac{\partial^2 u_\theta}{\partial x^2} - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right) = 0,
\end{aligned} \tag{3}$$

$$\frac{\partial u_x}{\partial x} + \frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0. \tag{4}$$

Therefore, in this situation the momentum equations are second order semi-linear partial differential equations. The continuity equation (4) is a first order linear partial differential equation. After non-dimensionalization (see Appendix C for the 3D example), setting $\rho = 1$, and setting the forcing function to 0 (no gravity), these equations, in coordinate-independent notation, become (see Cliffe *et al.* [16]):

$$\frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re}\nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{0}, \tag{5}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{6}$$

In the following subsections first the full 3D geometry and equations will be discussed, after which the symmetry reduced 2D geometry and equations will be discussed. For the results of the calculations, this order is reversed, because the 2D calculations were to easier and faster to perform.

1.3 3D case

In this case, the flow of an incompressible fluid is confined in a cylindrical pipe with the domain $\Omega \subset \mathbb{R}^3$. Here $\Omega = [0, 2\pi\rangle \times ([0, R_1] \times [0, L_1] \cup [0, R_2] \times [L_1, L_2])$. The radius of the inlet pipe is R_1, and the radius of the outlet pipe is R_2, with $R_1 < R_2$. In the numerical experiments in this report this ratio will be set equal to $1 : 2$. In a cylindrical coordinate

system, the non-dimensionalized, incompressible Navier-Stokes equations are [15], as also derived in Appendix C:

$$\begin{aligned}
\frac{\partial u_x}{\partial t} - \frac{1}{Re}\nabla^2 u_x + (\nabla u_x)\cdot\mathbf{u} + \frac{\partial p}{\partial x} &= 0, \\
\frac{\partial u_r}{\partial t} - \frac{1}{Re}\left(\nabla^2 u_r - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right) + (\nabla u_r)\cdot\mathbf{u} - \frac{u_\theta^2}{r} + \frac{\partial p}{\partial r} &= 0, \\
\frac{\partial u_\theta}{\partial t} - \frac{1}{Re}\left(\nabla^2 u_\theta - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right) + (\nabla u_\theta)\cdot\mathbf{u} + \frac{u_r u_\theta}{r} + \frac{1}{r}\frac{\partial p}{\partial \theta} &= 0,
\end{aligned} \tag{7}$$

$$\frac{\partial u_x}{\partial x} + \frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0. \tag{8}$$

The (3D, cylindrical) momentum equations for this situation are second order semi-linear partial differential equations. The continuity equation is a first order linear partial differential equation.

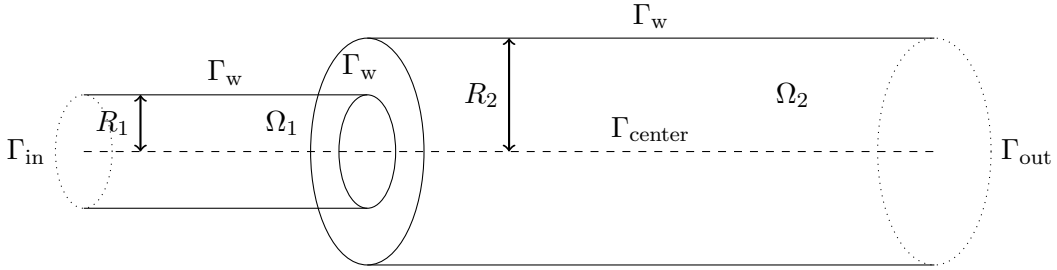

Figure 1: Walls of the cylinder, image adapted from an image in article by Cliffe *et al.* [15]. The proportions are not correct.

Due to transformation to and the use of cylindrical coordinates, a singularity at $r = 0$ is introduced. It will be treated as a weakly enforced boundary condition, similar to the 2D calculation. This will then disappear in the FEM (section 2.1)

Boundary Conditions The incoming flow (on Γ_{in}) is assumed to be Poiseuille:

$$u_x = u_{max}(1 - r^2), \tag{9}$$
$$u_\theta = 0, \tag{10}$$
$$u_r = 0. \tag{11}$$
$$\tag{12}$$

No-slip conditions are imposed on the wall (Γ_w):

$$u_x = u_r = u_\theta = 0. \tag{13}$$

Natural outflow conditions are assumed for outflow (Γ_{out}) [15](difference in notation due to different form of non-dimensionalisation):

$$\frac{1}{Re}\frac{\partial u_x}{\partial \mathbf{n}} - p n_x = \frac{1}{Re}\frac{\partial u_r}{\partial \mathbf{n}} - p n_r = \frac{1}{Re}\frac{\partial u_\theta}{\partial \mathbf{n}} - p n_\theta = 0. \tag{14}$$

The outlet of the second part of the pipe should be sufficiently long to allow the flow to be fully developed at the exit. Otherwise the outflow condition will not be met.
And finally, natural boundary conditions are assumed for the singularity at $r = 0$:

$$\frac{1}{Re}\frac{\partial u_x}{\partial n_x} - pn_x = \frac{1}{Re}\frac{\partial u_r}{\partial n_r} - pn_r = 0. \tag{15}$$

1.4 3D case reduced to 2D calculation

Since for these calculations the pipe (with Poiseuille inlet velocity, see boundary conditions) exhibits O(2) symmetry (see also Appendix G). The inlet velocity does not depend on θ and u_θ is assumed to be 0 (no rotation), the 3D problem for the pipe can be reduced to a 2D calculation. This geometry is shown schematically in Figure 2. The spatial domain reduces to $\Omega = (\Omega_1 \cup \Omega_2) = [0, R_1] \times [0, L_1] \cup [0, R_2] \times [L_1, L_2]$ and the equations reduce to:

$$\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t} + u_r\frac{\partial u_x}{\partial r} + u_x\frac{\partial u_x}{\partial x}\right) + \frac{\partial p}{\partial x} - \mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_x}{\partial r}\right) + \frac{\partial^2 u_x}{\partial x^2}\right) = 0,\\
\rho\left(\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + u_x\frac{\partial u_r}{\partial x}\right) + \frac{\partial p}{\partial r} - \mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{\partial^2 u_r}{\partial x^2} - \frac{u_r}{r^2}\right) = 0,\\
\frac{\partial u_x}{\partial x} + \frac{1}{r}\frac{\partial (r u_r)}{\partial r} = 0.
\end{aligned} \tag{16}$$

The radius of the inlet pipe is R_1, and the radius of the outlet pipe is R_2, with $R_1 < R_2$. In the numerical experiments in this report this ratio will be set equal to $1 : 2$.

A term $-\frac{1}{2}(\nabla \cdot \mathbf{u})\mathbf{u}$ to equation (5) was added for stability in the article by Cliffe *et al.* [15]. Even though another method (continuous Galerkin) than the one in the article (discontinuous Galerkin) was used, this term is included in AptoFEM for the 2D calculations, and therefore should be mentioned in this report. This is allowed, because the continuity equation is equal to 0.

The equations then become:

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re}\nabla^2\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \frac{1}{2}(\nabla\cdot\mathbf{u})\mathbf{u} = \mathbf{0},\\
\nabla\cdot\mathbf{u} = 0.
\end{aligned} \tag{17}$$

Figure 2 shows the layout of the half-channel for which the calculations were performed.

Boundary Conditions The incoming flow (on Γ_{in}, therefore at $x = 0$) is assumed to be Poiseuille (Dirichlet boundary condition). Therefore u_x versus r is a quadratic function.

$$\begin{aligned}
u_{x,\text{in}} &= u_{\max}(R_1^2 - r^2),\\
u_{r,\text{in}} &= 0.
\end{aligned} \tag{18}$$

No-slip conditions are imposed on the wall (Γ_{w}):

$$\mathbf{u} = 0. \tag{19}$$

Natural outflow conditions (Neumann boundary condition) are assumed at the exit [13]:

$$\frac{1}{Re}\frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n} = 0. \tag{20}$$

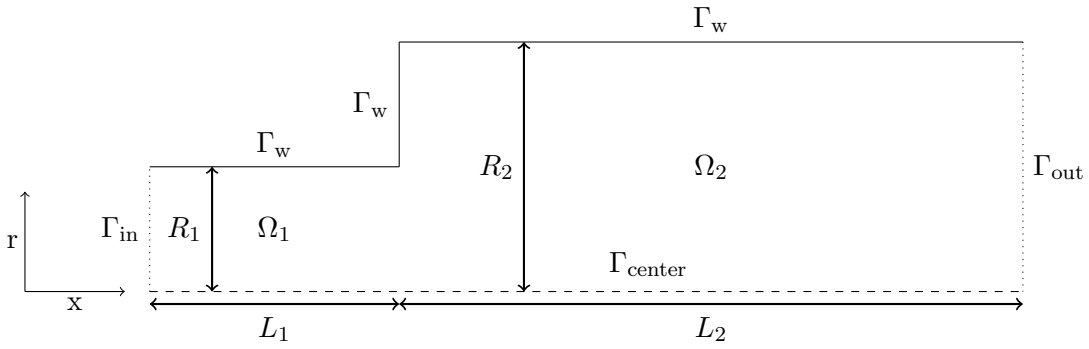

Figure 2: Walls of the channel, image adapted from an image in article by Cliffe *et al.* [16]. For clarity, the proportions are not correct.

Again, the outlet of the second part of the channel should be sufficiently long to allow the flow to be fully developed at the exit, otherwise the outflow conditions will not be met. And finally at the center of the channel (Γ_{center}) [13] $u_r = 0$ and:

$$\frac{1}{Re}\frac{\partial u_x}{\partial n_x} - pn_x = 0. \tag{21}$$

1.5 Literature

In the case of a laminar, incompressible fluid flow through a channel with a sudden expansion, an increasing Reynolds number eventually causes the reflectional symmetry around the central axis to be broken at a pitchfork bifurcation [20]. For the $1:3$ sudden expansion, the critical Reynolds number lies around 40 [20]. This was shown both in physical experiments and in numerical simulations. In this setup [13], the discontinuous Galerkin adaptive refinement algorithm, based on a posteriori error estimation, as opposed to the uniform refinement algorith, proved to be better: using the same number of degrees of freedom, the error in the eigenvalue is always smaller.

In the 3D-case, the flow of an incompressible fluid through a cylindrical pipe with an axisymmetric sudden expansion was considered by Mullin *et al.* [29]. They observed a flow asymmetry in physical experiments. The observation of asymmetry was estimated to be from critical Reynolds number of 1139 ± 10. This Reynolds number, based on the pipe diameter and average velocity (see Appendix A), points towards laminar flow.

This Reynolds number was found in (3D-) experiments performed by Mullin *et al.* [29], using an expansion ratio of 1:2. A sudden increase of the measure for asymmetry squared was observed. In the experiment the flow was measured using high resolution magnetic resonance imaging, measuring the orientation of the spins of the hydrogen protons of the water after spin flip and precessing. The plot of the measure of asymmetry squared versus the Reynolds number showed a typical pattern for a Hopf or pitchfork bifurcation, at $Re = 1139 \pm 10$, see Figure 3. However, further theoretical investigations on the bifurcation, by Cliffe *et al.* [15], have shown no evidence of such a bifurcation at the given range of Reynolds numbers. At the present, it is therefore assumed a phenomena, different from the known bifurcations, causes this pattern. At higher Reynolds numbers (above approximately 1550), Mullin *et al.* [29] observed a time dependence of the solutions found. The steady asymmetric flow is stable until the flow becomes time dependent, at a

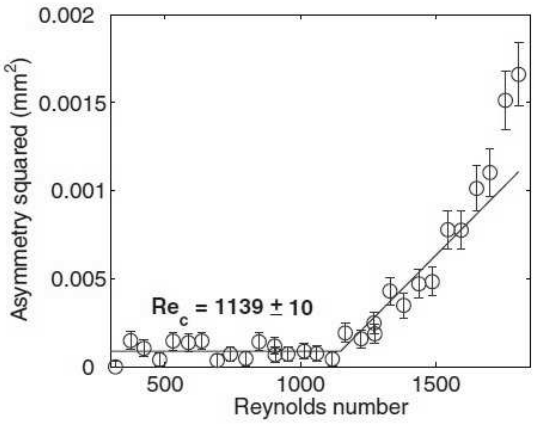

Figure 3: Reynolds number versus measure of asymmetry squared, graph from article by Mullin *et al.* [29].

Reynolds number of 1453±41 [29]. Cliffe *et al.* [14], using an *hp*-adaptive algorithm, found a critical Reynolds number of around 5000. In a further article [15], they investigated the dependence of the critical Reynolds number on the value of R_2, the radius of the right part of the pipe. For $R_2 = 2.6$ (and $R_1 = 1.0$) they found an overall minimum critical Reynolds number of around $Re \approx 4000$.

For the 2D flow through an expanding channel, Biswas *et al.* [6] found that increasing the Reynolds number leads to an increase in the side wall effect (caused by no-slip boundaries), and thereby an increase in reattachment length for the vortex.

In a similar setup, flow through an axisymmetric stenosis, with small asymmetries, Griffith *et al.* [24] investigated the flow for Reynolds numbers up to 400. The Reynolds number was defined using the average velocity and the diameter. In experimental results, the critical Reynolds number for transition to an unstable flow was found to be lower than predicted by numerical experiments. It has been suggested that those perturbations from experimental results might be caused by small non-axisymmetric imperfections in the experiment [35]. An indication of this phenomena was also shown by Sanmiguel-Rojas and Mullin [33]. Griffith *et al.* [24] have shown the patterns for these stenoses with small offsets show similar patterns as those observed in the experiments.

Cantwell *et al.* [10] have shown that the the flow through a pipe with a sudden expansion is highly sensitive to disturbances in the velocity inlet profile, but only transiently.

1.6 Goal

An asymmetric flow pattern was observed for Reynolds numbers greater than 1139±10 in practical experiments by Mullin *et al.* [29] for the axisymmetrical expansion of a cylindrical pipe, but this was not observed in the accurate numerical calculations.

The goal is to implement a new and robust linear equations solver into the AptoFEM software for a mixed finite element/spectral discretisation of the problem. At first we will look at the flow of a viscous fluid through a symmetrical pipe with a sudden expansion (symmetry reduced 2D representation of the 3D problem), we will then extend this (3D) with dependence on θ.

2 Linear Solver and discretisation

2.1 Finite Element Methods (FEM)

Finite element methods are used in numerical mathematics to approximate the solution to a boundary value problem. The following subsection was derived from lecture notes by E.J.C. Hall [25]. The ith equation of a general second order semilinear differential system, defined in the bounded domain Ω, is given below:

$$\sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial^2 u_i(\mathbf{x})}{\partial x_j^2} = f_i(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}). \tag{22}$$

Here $\mathbf{u} \in \mathrm{R}^{N_i}$ is the velocity vector, $\mathbf{x} \in \Omega$ the position vector, $a(\mathbf{x})$ a coefficient function depending on $\mathbf{x}$, and $\mathbf{f}$ vector of a general function of $\mathbf{u}$, $\nabla \mathbf{u}$ and $\mathbf{x}$; $i = 1, ..., N_i$, with N_i being the length of the position and velocity vectors and $j = 1, ..., N_i$. $\Omega \in \mathbb{R}^{N_i}$, with $N_i \geq 1$, $f_i \in C(\bar{\Omega}) \forall i \in [1, 2, ..., N_i]$.

The first step in solving system of equations with the finite element methods, is to rewrite the PDE into its weak/variational form. This is achieved by multiplying each equation in the system with a test function, followed by integration over the domain. For now we assume that the test function for the ith equation is given by $v_i(\mathbf{x}) \in \mathbb{R}^{N_i}$.

$$\sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial^2 u_i(\mathbf{x})}{\partial x_j^2} v_i(\mathbf{x}) = f_i(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}) v_i(\mathbf{x}) \tag{23}$$

The resulting equation after multiplication is integrated over the domain Ω.

$$\int_\Omega \sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial^2 u_i(\mathbf{x})}{\partial x_j^2} v_i(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_\Omega f_i(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}) v_i(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{24}$$

Partial integration then gives:

$$\int_{\partial\Omega} \sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial u_i(\mathbf{x})}{\partial x_j} n_j v_i \mathrm{d}\mathbf{s} - \int_\Omega \sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial u_i(\mathbf{x})}{\partial x_j} \frac{\partial v_i(\mathbf{x})}{\partial x_j} \mathrm{d}\mathbf{x} = \int_\Omega f_i(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}) v_i(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{25}$$

Here $\delta\Omega$ indicates the boundary of Ω, and $\mathbf{n}$ is the outward pointing normal. The weak formulation is obtained by choosing the test function such that $v_i|_{\partial\Omega} = 0$.

The new problem will be:

Find a $\mathbf{u}$ such that $\mathbf{u}|_{\partial\Omega} = 0$ and

$$\int_\Omega \sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial u_i(\mathbf{x})}{\partial x_j} \frac{\partial v_i(\mathbf{x})}{\partial x_j} \mathrm{d}\mathbf{x} = -\int_\Omega f_i(\mathbf{u}, \nabla \mathbf{u}, \mathbf{x}) v_i(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{26}$$

for all v_i such that $v_i|_{\partial\Omega} = 0$.

Introducing some formal definitions:

$L_2(\Omega)$ is the set of all real-valued functions defined on a open subset $\Omega \subset \mathbb{R}^n$ such that:

$$\|u\|_{L_2(\Omega)} \equiv \left(\int_\Omega |u(x)|^2 \mathrm{d}x \right)^{\frac{1}{2}} < \infty. \tag{27}$$

The first order Sobolev space is defined as:

$$H^1(\Omega) \equiv \{u \in L_2(\Omega) : D^1 u \in L_2(\Omega)\}. \tag{28}$$

Here D indicates the weak derivative. However, because u is assumed to be at least C^1 continuous, this can be replaced by the first order partial derivative.

$$H_0^1(\Omega) \equiv \{v \in H^1(\Omega) : v(\partial\Omega) = 0\} \tag{29}$$

$v(\partial\Omega) = 0$ indicates v should be 0 on the boundaries of Ω.

Therefore the problem can be reformulated to: Find a $\mathbf{u} \in H_0^1(\Omega)^{N_i}$ such that

$$\int_\Omega \sum_{j=1}^{N_i} a(\mathbf{x}) \frac{\partial u_i(\mathbf{x})}{\partial x_j} \frac{\partial v_i(\mathbf{x})}{\partial x_j} \mathrm{d}\mathbf{x} = \int_\Omega f_i(\mathbf{u}, \nabla\mathbf{u}, \mathbf{x}) v_i(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{30}$$

for all $v_i \in H_0^1(\Omega)^{N_i}$.

The second step consists using a finite dimensional subset of the function space to solve the equation. Following this, the system is discretized, by dividing the interval into subintervals (elements), resulting in the mesh. Basis functions are defined on the mesh. The basis function $\phi_i(\mathbf{x})$ have a value of 1 at the point i, and a value of 0 on the edges of element i, and outside element i. Because these basis functions are defined this way, the interaction between most mesh points is zero, resulting in a sparse matrix.

2.1.1 Weak form Navier-Stokes equations

The weak form of the stationary Navier-Stokes equations, is given by:

$$\begin{aligned} \int_\Omega (\mathbf{u}\nabla\mathbf{u}) \cdot \mathbf{v} - p \cdot (\nabla\mathbf{v}) + \frac{1}{Re}(\nabla\mathbf{u}) : (\nabla\mathbf{v}) \mathrm{d}\mathbf{x} = 0, \\ \int_\Omega q \nabla \cdot \mathbf{u} \mathrm{d}\mathbf{x} = 0. \end{aligned} \tag{31}$$

For the problem in this research, the new formulation will be:
Find a $\mathbf{u} \in H_E^1(\Omega)$ and $p \in L_2(\Omega)$ such that:

$$\begin{aligned} \int_\Omega (\mathbf{u}\nabla\mathbf{u}) \cdot \mathbf{v} - p \cdot (\nabla\mathbf{v}) + \frac{1}{Re}(\nabla\mathbf{u}) : (\nabla\mathbf{v}) \mathrm{d}\mathbf{x} \\ - \int_{\partial\Omega} \frac{1}{Re}(\nabla\mathbf{u} \cdot \mathbf{n})\mathbf{v} \mathrm{d}\mathbf{s} + \int_{\partial\Omega} (p\mathbf{n}) \cdot \mathbf{v} \mathrm{d}\mathbf{s} = \\ \int_\Omega (\mathbf{u}\nabla\mathbf{u}) \cdot \mathbf{v} - p \cdot (\nabla\mathbf{v}) + \frac{1}{Re}(\nabla\mathbf{u}) : (\nabla\mathbf{v}) \mathrm{d}\mathbf{x} \\ - \int_{\Gamma_{\text{out}}} (\frac{1}{Re}\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - p\mathbf{n})\mathbf{v} \mathrm{d}\mathbf{s} - \int_{\Gamma_{\text{center}}} (\frac{1}{Re}\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - p\mathbf{n})\mathbf{v} \mathrm{d}\mathbf{s} = 0, \\ \int_\Omega q \nabla \cdot \mathbf{u} \mathrm{d}\mathbf{x} = 0. \end{aligned} \tag{32}$$

for all $\mathbf{v} \in H_{E0}^1(\Omega)$ and for all $q \in L_2(\Omega)$. Here, like in the book by Elman *et al.* [18, pp. 222]

$$H_{E0}^1(\Omega) \equiv \{v \in H^1(\Omega)^3 : \mathbf{v}(\partial\Omega) = \mathbf{0}\}, \tag{33}$$

$$H_E^1(\Omega \equiv \{\mathbf{u} \in H^1(\Omega)^3 : \mathbf{u}(\Gamma_{\text{wall}}) = \mathbf{0} \wedge u_x(\Gamma_{\text{in}}) = u_{\max}(R_1^2 - r^2) \wedge u_r(\Gamma_{\text{in}}) = u_\theta(\Gamma_{\text{in}}) = 0\}. \tag{34}$$

By finding basis functions v and q for which the function is 0 on and outside the edges of an element and 1 in its interior, the local matrix can eventually be constructed. This will not further be specified as it is laborious and does not add much to the problem.

After discretisation this yields $\mathbf{U} \in V_h^3$ and $P \in W_h$, where $V_h^3 \times W_h$ is the FEM function space of piecewise polynomials originating from $H_{bc}^1(\Omega)^3 \times L_2(\Omega)$.

The seemingly natural basis with equal polynomial degrees for the pressure and velocity is not stable, as described in the book by Elman *et al.* [18, pp. 228-233], 'care must be taken to ensure the velocity space is rich enough compared to the pressure space'. The inf-sup stability is achieved by satisfying the inf-sup condition [18] and thereby also satisfying the discrete solvability condition. The inf-sup condition is given by [18]:

$$\min_{q_h \neq cst} \max_{\mathbf{v}_h \neq 0} \frac{|(q_h, \nabla \mathbf{v_h})|}{||\mathbf{v_h}||_{1,\Omega} ||q_h||_{0,\Omega}} \geq \gamma. \tag{35}$$

Here γ is a positive constant, independent of the mesh size h, $\mathbf{v}_h \in V_j$ and $q_h \in W_h$. For a 2D-quadrilateral mesh, a stable approximation would for instance be velocity having polynomial degree $p = 2$, and pressure $p = 1$, also called the Taylor-Hood method, [18, p. 229]. Elman *et al.* claim the Taylor-Hood method is uniformly stable for every possible grid with more than one element [18, p. 233], which they showed using macroelements.

2.2 Spectral methods

In spectral methods, the solution of a differential equation is written as a sum of basis functions [9]. Spectral methods use global basis functions, whereas FEM use local basis functions. For a periodic function, a Fourier expansion can best be used in spectral discretisation, as these basis functions automatically satisfy the periodicity. The spectral discretisation has an advantage of a higher accuracy over the FEM when the solution is smooth. Therefore a lower number of frequencies, N, is required [11].

In general the polynomial degree of the spectral basis functions is high, and the functions are non-zero in the domain, except for at certain isolated points [9]. Because of the non-zero basis functions, spectral discretisation results in full matrices, and therefore the matrix systems are generally more difficult and more expensive to solve than the FEM (sparse) matrix systems. The 3D combination of the spectral discretisation (θ-direction) and FEM (x,r-directions) in this research results in an extension to a sparse matrix with block matrices of the elements in the matrix in the 2D calculation. The sparse blocks are caused by the way the variables and mesh are ordered in the software.

Spectral methods are often implemented by either collocation or Galerkin methods. In the collocation method, the solution is prescribed in a certain number of points. The residual at those points should be equal to 0. For the Galerkin methods, the PDE should be written in its weak form. The resulting test functions, v_k, are assumed to be equal to the trial functions ($\phi_i(\theta)$) obtained by the (N-th order) approximation of the solution:

$$u(x, r, \theta) \approx u^N(x, r, \theta) = \sum_{i=0}^{N} \hat{u}_i(x, r)\phi_i(\theta). \tag{36}$$

It is often easier to derive a solution for the collocation method than for the Galerkin methods. The collocation method will be used in this research for discretisation in the θ-direction.

The error estimate of the combined spectral discretisation and FEM depends on the mesh size h and the largest wave number N [12]. In order to obtain a smooth solution, N^{-1} should be chosen asymptotically much larger than h, the step size in the x- or r-direction [11].

2.2.1 Fourier spectral method

Since the velocity is assumed to be 2π-periodic in the θ-direction, modelling the system via a combination of the spectral method in the θ-direction and continuous Galerkin (CG) finite element methods in the x- and r-direction might be a good approximation. Continuous Galerkin briefly means the approximation of the function at the boundaries of neighbouring elements is continuous. This model will result in a discrete solution for the velocity which is a trigonometric polynomial in the θ-direction and a piecewise polynomial in the x- and r-directions [11].

Because of a discontinuity in the r-direction (jump from R_1 to R_2), a spectral discretisation in the r-direction would not be suitable [30], due to the Gibbs phenomenon. (Even though the order of the approximation is increased, and the functions approach the discontinuity, at the discontinuity the approximation does not improve.) Since the r- and x-direction do not exhibit the periodicity that the θ-direction displays, this does not display the same advantage of spectral methods over the FEM-methods.

The Fourier spectral discretized expansion of the velocity is given by:

$$u(x, r, \theta) \approx \sum_{i=0}^{N} \hat{u}_i(x, r) \exp(\theta_i). \tag{37}$$

The derivation of the elements in the spectral collocation matrix can be found in Appendix D. The θ-discretisation of the Navier-Stokes equations can be found in Appendix E. The Python code for generating the Fortran90 code can be found in Appendix F.

2.3 Damped Newton method

The damped Newton method is used as a solver to the nonlinear system in AptoFEM, the main software used. If the initial guess is too close to a singularity to converge to a solution, the method uses damping, this may result in a solution being found. A diagonal correction term, ω, is added to the stiffness matrix [21].

Given the system $F(U) = 0$, with U the solution vector, the Newton method works as follows [1]: U^0 is the initial guess.

Using an inner solver (in this case MUMPS or Ilupack, possibly combined with the spectral method), the system $J(U^n)\delta U^{n+1} = -F(U^n)$ is solved for δU^{n+1}, where J is the Jacobian matrix. The solution is updated to:

$$U^{n+1} = U^n + \omega \delta U^n, \tag{38}$$

where ω is the damping factor; $0 < \omega < 1$. $\omega = 1$ indicates no damping. These steps are repeated until the error $(F(U^{n+1}) - F(U) = F(U^{n+1}))$is smaller than the tolerance. The Newton method shows a quadratic convergence (in the case of no damping).

$J\delta U^n = -F(U^n)$ is discretized using the finite element methods.

The residual is determined as $\delta U^{n+1} = U^{n+1} - U^n$ (with $n \in \mathbb{N}$). The algorithm would look somewhat like:

$F(U) = 0$
$It_{nr} = 0$
Error $= F(U^{n+1}) - F(U) = F(U^{n+1})$
while Error $<$ tolerance **do**
  while $It_{nr} \leq It_{max}$ **do**
    $\delta U^n = -J(U^n)^{-1} F(U^n)$
    $U^{n+1} = U^n + \omega \delta U^n$
    $It_{nr} = It_{nr} + 1$
  end while
end while.

Here It_{nr} is an integer which counts the number of Newton iterations performed, allowing for a maximum number of Newton iterations.

2.4 Continuation in the Reynolds number

For the continuation in the Reynolds number used in this research, the solution to a previous calculation, $\mathbf{W}(Re_{n-1})$, was used as an initial guess for the solution in the new calculation, $\mathbf{W}(Re_n)$.

Construct $\mathbf{W}$(Re) s.t.
$F(\mathbf{W}, Re) = 0$
$\mathbf{W}(Re_n) = \mathbf{W}(Re_{n-1}) + \Delta Re \frac{\partial \mathbf{W}}{\partial Re}$
$\frac{\partial \mathbf{W}}{\partial Re}(Re_{n-1}) \approx \frac{\mathbf{W}(Re_{n-1}) - \mathbf{W}(Re_{n-2})}{Re_{n-1} - Re_{n-2}}$

In this case $F(\mathbf{W}, Re)$ are the Navier-Stokes equations, and $\mathbf{W}$ is a vector containing $\mathbf{U}$ and p.

2.5 Discretisation

2.5.1 Mesh

In the case of reduction to 2D, the domain was divided into quadrilaterals. They should have interior angles which are smaller than 180°.

In the spectral method case, the (x,r)-domain was also divided into quadrilaterals. For the θ-direction, a spectral discretisation should be used. The periodicity of θ is preserved, due to the use of the Fourier basis functions, see also Appendix E.

2.5.2 Degrees of freedom

The number of degrees of freedom can then be determined as follows:

Each nodal point contains a u_x, u_r and p term. Every edge and every interior point contains 2 extra terms (from u_x and u_r). Therefore an estimate for the number of degrees of freedom in the 2D calculation can be calculated by:

$$no_{\text{dof}} = 3 \times no_{\text{nodes}} + 2 \times no_{\text{edges}} + 2 \times no_{\text{interior points}}. \tag{39}$$

The boundary conditions limit the actual number of degrees of freedom. For instance, an inflow boundary condition will determine the value of the velocity, and therefore the

number of degrees of freedom decreases by $2 * no_{\text{nodes at x=0, r}} + 2 * no_{\text{edges at x=0, r}}$. For a suitably large mesh size, this will not decrease the number of degrees of freedom significantly.

For the full 3D calculation, the number of degrees of freedom can be calculated by:

$$no_{\text{dof}} = 4 \times no_{\text{nodes}} + 3 \times no_{\text{edges}} + 3 \times no_{\text{interior points}}. \tag{40}$$

2.5.3 Resulting matrix

In both the 3D flow through a pipe with sudden expansion and the 3D flow through a pipe with sudden expansion, reduced to a 2D calculation, the system can be described as (see also Figures 4, 5 and 6):

$$\begin{pmatrix} K & C \\ C^\top & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ P \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}. \tag{41}$$

Here K is the contribution of $\mathbf{U}$ to the Navier-Stokes equations. $\mathbf{0}$ is caused by the incompressibility constraint: the continuity equation does not depend on P. $\mathbf{U}$ and P are the discretized versions of the velocity and pressure respectively. The 3D-system has $O(2)$ symmetry [15] (rotation and reflection), within the azimuthal direction of the cylindrical pipe, for the non-discretized case with Poiseuille inlet conditions, see Appendix G.

3 Solvers

3.1 AptoFEM

AptoFEM is a software package for solving systems of partial differential equations, using FEM. It interfaces MUMPS as a direct solver method. Ilupack was implemented into AptoFEM for this research. The most relevant input variables for this research were the mesh type, mesh size, Reynolds number, the size of the object, the problem dimension and polynomial dimension. GMRES is used as an preconditioned iterative solver. The Reynolds number in AptoFEM is based on the definition with the radius and maximum velocity. The maximum inlet velocity in the length of the channel is initially set equal to 1, for the 3D calculations a perturbed inlet velocity is also used.

3.2 MUMPS: a direct solver

MUMPS is an abbreviation for MUltifrontal Massively Parallel Sparse direct solver. It is based on the LU-factorization of the matrix [4]. It can solve a large, linear systems of equations with sparse matrices, which can be either symmetric, unsymmetric or symmetric positive definite [2]. It can perform several computations simultaneously (parallel method). However, for all the calculations performed in this research project, the serial version of MUMPS was used.

The first step in solving a linear system using MUMPS, is the preprocessing of the matrix, and LU-factorization. The preprocessing consists of numerical pivoting, trying to get the largest entries in a row on or near the diagonal, using symmetric permutations and scaling matrices [22]. An approximation of the solution, $\hat{\mathbf{x}}$, to the thus obtained system $A\mathbf{x} = \mathbf{b}$ is then calculated. Its residual, **res**, is determined, and if necessary, $A\delta\mathbf{x} = \text{res}$ is calculated by iterative refinement, and the updated solution will be $\hat{\mathbf{x}}_{new} = \hat{\mathbf{x}} + \delta\hat{\mathbf{x}}$ [22].

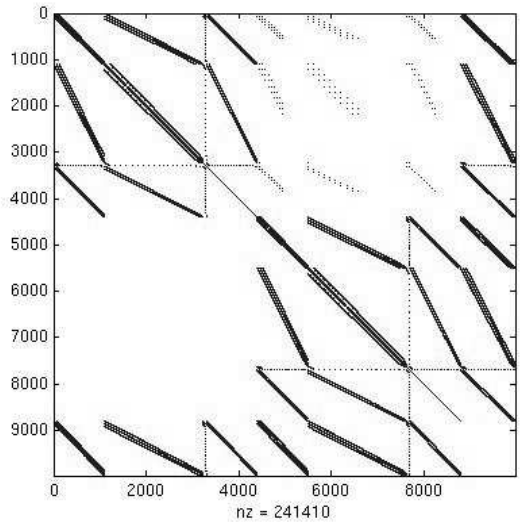

Figure 4: Matrix visualization by Matlab spy for the matrix obtained from AptoFEM, solved using Ilupack, for $Re = 50$, with grid (11 51; 11 11) for the 1:2 expansion, with channel length 50, using the initial guess 0 (no Newton iterations performed yet).

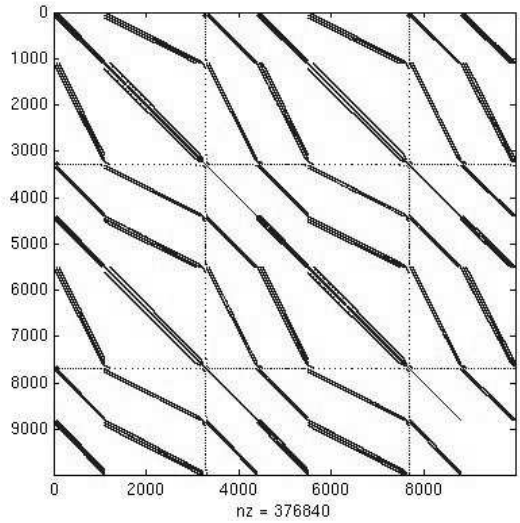

Figure 5: Matrix visualization by Matlab spy for the matrix obtained from AptoFEM, solved using Ilupack for $Re = 50$, with grid (11 51; 11 11) for the 1:2 expansion, with channel length 50, after 2 Newton iterations.

($\delta\mathbf{x}$ denotes a small variation with respect to $\hat{\mathbf{x}}$.) This is repeated until the required accuracy is achieved. The iterative refinement option is not used in this research, and since MUMPS is a direct solver, MUMPS converged within one step. Near bifurcation points the matrix becomes singular. The resulting solution may be more affected by finite precision errors. In those cases it is wise to use iterative refinement.

MUMPS is the direct linear solver used in inner iterations in AptoFEM and it was

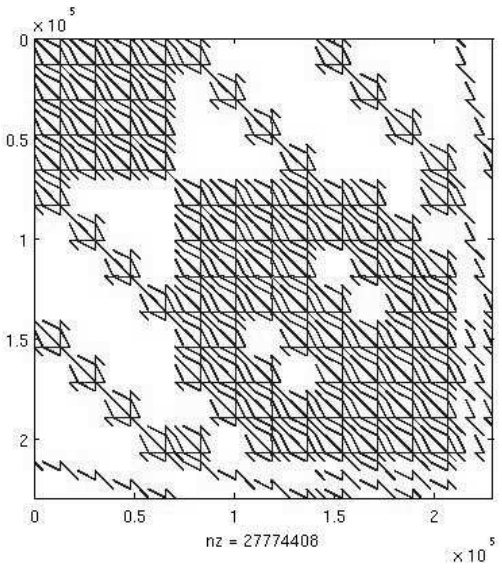

Figure 6: Matrix visualization by Matlab spy for the matrix obtained from AptoFEM, for $Re = 10$, with grid (21 101; 21 21) for the 1:2 expansion, with channel length 50, after 2 Newton iterations, and for the 3D calculation with 4 collocation points and symmetric inlet velocity

used by Cliffe *et al.* [13] for modelling the (2D) flow through the (1:3) sudden expansion of a channel. (Damped Newton iterations were used for the outer iterations.) MUMPS can run in parallel [3, 5], which enables faster calculation.

3.3 Ilupack: an iterative solver

Ilupack is an iterative solver, based on incomplete LU factorisations. It can solve real and complex systems, symmetry is not required. In this research Ilupack was implemented in AptoFEM as another optional solver. Ilupack is based on Krylov subspaces methods.

3.3.1 Krylov subspaces methods

The spanning vectors of a Krylov subspace are [36]:

$$\mathcal{K}^m(A,\ bfv) \equiv \text{span}\{\mathbf{v}, A\mathbf{v}, ..., A^{m-1}\mathbf{v}\}. \tag{42}$$

For increasing power of A, the vectors become increasingly similar and therefore increasingly linear dependent [36]. For the Krylov subspace method the vectors $\mathbf{v}$, from equation (42), are equal to: $\mathbf{v} = \mathbf{b} - A\mathbf{x}_0$ [31, p.151]. The approximation is then of the form:

$$A^{-1}\mathbf{b} \approx \mathbf{x}_m = \mathbf{x}_0 + \mathbf{q}_{m-1}(A)\mathbf{v}. \tag{43}$$

Where q_{m-1} is a polynomial of degree m-1 [31].

3.3.2 Generalized Minimal Residual Method (GMRES)

GMRES is an iterative method for numerically solving a system of linear equations $A\mathbf{x} = \mathbf{b}$. The solution is approximated by $\mathbf{x}_n \in \mathcal{K}^n(A, \mathbf{b})$, which minimizes the Euclidean norm of $A\mathbf{x}_n - \mathbf{b}$. This is calculated for all vectors in the basis of the Krylov space.

The speed of convergence of the Krylov power method depends on the ratio of the eigenvalues λ_1/λ_2, where λ_i indicates the i^{th}-largest eigenvalue [36]. A great advantage of GMRES over many other iterative methods for solving linear systems is that it cannot break down due to properties of the matrix until it has converged [32].

To overcome the vectors in the Krylov subspace becoming increasingly similar, the Arnoldi method is used to find a more stable basis for the Krylov subspace, resulting in an orthonormal basis, consisting $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n$ column vectors in Q_n.

The Arnoldi iteration work as follows: start with a normalized, real vector $\mathbf{q}_1 = \frac{\mathbf{b}}{||\mathbf{b}||}$ [19].

for $n = 1, 2, ..n-1$ **do**
  $\mathbf{v} = A\mathbf{q}_n$
  for $j = 1, 2, ..., n$ **do**
    $h_{jn} = \mathbf{q}_j\mathbf{v}$
    $\mathbf{v} = \mathbf{v} - h_{jn}\mathbf{q}_j$
  end for

  $h_{n+1,n} = ||\mathbf{v}||_2$
  $\mathbf{q}_{n+1} = \frac{\mathbf{v}}{h_{n+1,n}}$
end for

For GMRES, the extra iteration steps are [19]:

for $n = 1, 2, ..n-1$ **do**
  $\min_{\mathbf{y}} ||H_n\mathbf{y} - ||\mathbf{b}||\mathbf{e_1}||_2$
  $x_n = Q_n\mathbf{y}$
end for

where H_n is an upper Hessenberg matrix (upper triangular plus one extra subdiagonal below the diagonal line).

GMRES requires $(m+3+\frac{1}{m})N + NZ$ multiplication operations and $(m+2)N$ storage operations; here NZ is the number of nonzero elements in the matrix, N is the number of rows/columns of the matrix, m is the number of steps after which the method will be restarted [32]. For the calculations with Ilupack in this research, it is set to 30.

3.3.3 Preconditioning

A matrix K, for which $K^{-1}A$ has a better eigenvalue distribution, can be used for (left) preconditioning [36]. The spectrum of the matrix decreases, thereby improving convergence [34]. The system $K^{-1}A\mathbf{x} = K^{-1}\mathbf{b}$ should then be easier to solve [36]. The system $A\mathbf{x} = \mathbf{b}$ is multiplied by a matrix K^{-1}, and this is further applied to the Krylov subspace method [36]. If K^{-1} would be equal to A, the system would be solved in one iteration, in general this will be computationally to expensive. Therefore a factor K^{-1}, which is cheap to apply, is sought [36].

Preconditioning can for instance be done by (incomplete) LU-decomposition: the matrix is split up into a lower triangular matrix, an upper triangular matrix and a residual. In the incomplete LU decomposition, the residual terms are ignored.

Preconditioning in general lowers the condition number of the matrix (or on the distribution of eigenvalues), which is a measure of a systems sensitivity [37]. It can be defined

as [37]:

$$\kappa(A) = ||A||_2||A^{-1}||_2. \tag{44}$$

3.3.4 Ilupack

In this research Ilupack [8], a preconditioning software package, based on ILU-decomposition (incomplete LU-decomposition) of the matrix [7], was used for solving the Navier-Stokes equations. Ilupack uses GMRES for solving the system. The block matrix system is decomposed into an approximately LDU-form: respectively the block lower triangular, the block diagonal and the block upper triangular matrix. However the resulting lower diagonal block in the diagonal matrix is not truly diagonal, but a so-called Schur-complement. The matrix decomposition is then repeated for the Schur-block, until the drop tolerance $||E||_k < \epsilon$ is met [8]. The kth step of ILU-decomposition takes the following form [8]:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I \end{pmatrix} \begin{pmatrix} D_{11} & 0 \\ 0 & Sc_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & I \end{pmatrix} + E_k = L_k D_k U_k + E_k. \tag{45}$$

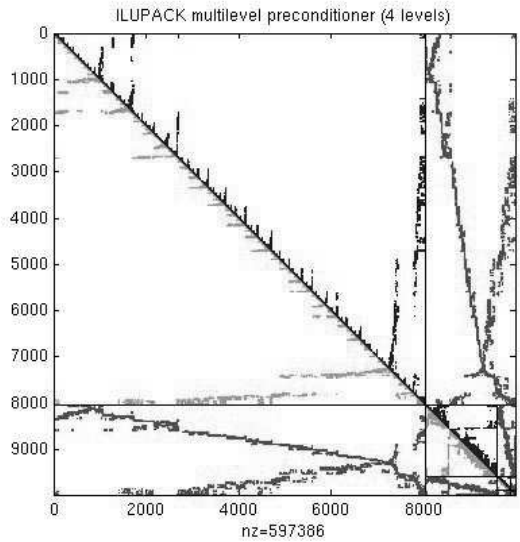

Figure 7: Matrix visualization by Matlab spy for the matrix obtained from AptoFEM, after preprocessing and reordering by Ilupack, for $Re = 50$, with grid (11 51; 11 11) for the 1:2 expansion, with channel length 50, with initial guess 0, no nonlinear terms included yet.

The matrices considered in this research are general real matrices. They are generally not symmetric and/or positive definite. Therefore, a routine for general real matrices from Ilupack was used: the DGNL-routine. (Here D implies double precision.) Moreover, since the matrices are nicely ordered, we do not use the matching routine from Ilupack for preprocessing. Ilupack in AptoFEM was compiled using double precision.

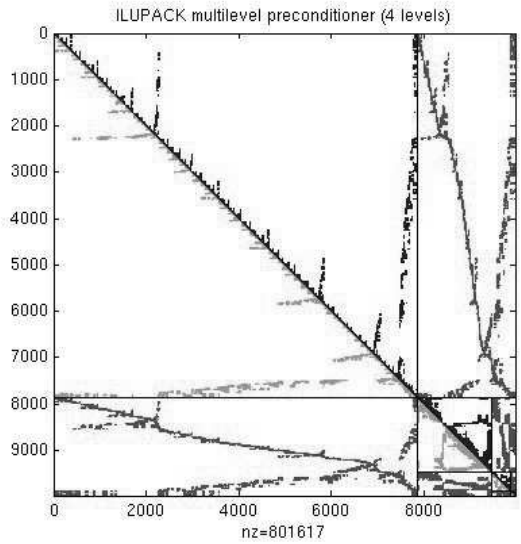

Figure 8: Matrix visualization by Matlab spy for the matrix obtained from AptoFEM, after preprocessing and reordering by Ilupack, for $Re = 50$, with grid (11 51; 11 11) for the 1:2 expansion, with channel length 50, after 2 Newton iterations.

4 Results

When graded, the mesh is graded in the x-direction, with initial width 0.1. A polynomial degree of 2 was used for velocities and a polynomial degree of 1 for pressures. The Newton tolerance was set to 1e-9 unless stated otherwise. The maximum number of Newton iterations was set to 100. The standard internal parameters were chosen for Ilupack unless mentioned otherwise. Calculations were performed for the channel with a 1:2 sudden expansion, and length of 10:50, unless stated otherwise. The matrix type was CSR (Compressed Sparse Row) for the 2D calculations, and mumps_unsymmetric for the 3D calculations. All the calculation times mentioned are CPU-times.

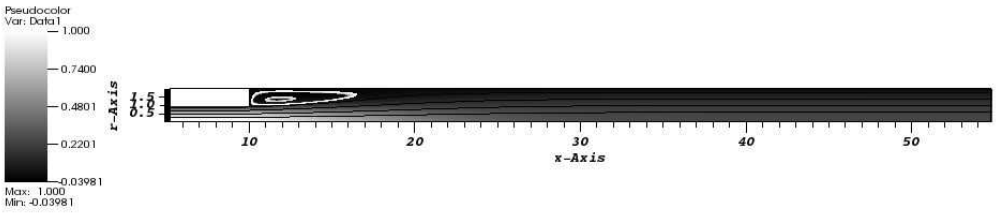

Figure 9: The 1:2 sudden expansion of a channel, with a velocity p2 and pressure p1 approximation, Re=100 and channel length 10:50. Numerical approximation was performed for the half-channel using Ilupack solver and (11, 11; 51, 11) grid.

Table 1: Amount of work and number of nonzeros in L for various orderings, from Bifurcation analysis lecture notes [37]

Numbering	flops/1000				nnz(L)/1000			
N	100	400	1600	6400	100	400	1600	1600
Random	35	968	78944	4477865	1.5	14	216	3110
Lex. graphical	11	165	2603	41302	1.0	8	64	512
Rev. Cuthill-McKee	7	96	1410	21510	0.8	6	45	351
Checkerboard	6	85	1299	20559	0.7	5	36	270
Nested dissection	7	78	804	7637	0.8	5	28	153
Minimum degree	5	53	590	7337	0.7	4	22	126

4.1 MUMPS vs Ilupack symmetry reduced 2D calculations

The 2D calculations were performed on Chiron, a Dell PowerEdge R410 with 2 x Xeon X5650 6-core processors (24 logical cores total) and 96 GB RAM. All executables were compiled using the -g flag (slower than the -03 flag).

Direct methods (*e.g.* MUMPS) are generally faster than iterative methods (*e.g.* Krylov subspaces in Ilupack) for 2D systems from 10.000 up to 100.000 unknowns, for 3D, the equilibrium is reached for smaller numbers of unknowns [37]. This is also observed for the calculations performed in this research (compare for instance the solution times in table 5 with those in table 6).

Ilupack uses the approximate minimum degree ordering, which is based on the fact that the row with the lowest number of elements should be picked as the pivot row [37] for factorisation. By choosing the row with the lowest number of elements, the least amount of fill-in will be created. The fill-in consists of the nonzero elements in the L and U matrices, which were not present in the original matrix [37]. As can be seen from Table 1, the minimum degree ordering requires the least amount of work from the orderings listed. For the (11, 51; 11, 11) grid with a Reynolds number of 50, the number of nonzeros increased from 376840 for the original, 5^{th} Newton iteration matrix to 771639 for the Ilupack preconditioned matrix. The fill-in consists of 394799 elements in this case. Within MUMPS the default is an automatic choice of the software between the ordering packages installed [2].

The complexity is the way in which the amount of work depends on the number of unknowns and therefore it depends on the ordering (and fill) of the matrix [37]. The work is influenced by the way the matrix elements are reordered. The complexity is optimal (minimum), when it depends linearly upon the number of unknowns [37], since the amount of work is equal to the number of iterations times the work per step. The number of iterations is approximately constant, and the work per step depends linearly on the number of unknowns [37].

For high dimensional matrices the direct method LU factorisation takes about $\frac{2}{3}n^3$ flops [37]. When the grid is refined, the number of iterations goes up. In Table 2, n is the number of rows of the matrix and N is the order of the matrix (the number of rows multiplied with the number of columns). In general the complexity of a method increases strongly with increasing dimension of a problem, which is illustrated for the nested dissection in Table 2. This holds for general other PDE-based problems as well.

In general the solution time of $LU\mathbf{x} = \mathbf{b}$ is proportional to storage and therefore cheap

Table 2: Complexity of nested dissection on Poisson problem on a hypercube, from Bifurcation analysis lecture notes [37]

	1D		2D		3D		dD	
factorisation	n	N	n^3	$N\sqrt{N}$	n^6	N^2	$n^{3(d-1)}$	$N^{\frac{3(d-1)}{d}}$
storage	n	N	$n^2 \log_2(n)$	$N \log_2(N)$	n^4	$N^{\frac{4}{3}}$	$n^{2(d-1)}$	$N^{\frac{2(d-1)}{d}}$

compared to factorisation [37, p. 108]. The Ilupack factorisation time is higher than the Ilupack solving time. For a larger grid size, the ratio of Ilupack factorisation time to solver time decreases.

From the obtained results, it can be observed that an increasing Reynolds number results in a larger recirculation zone, see also Figure 10 (visualisation with VisIt).

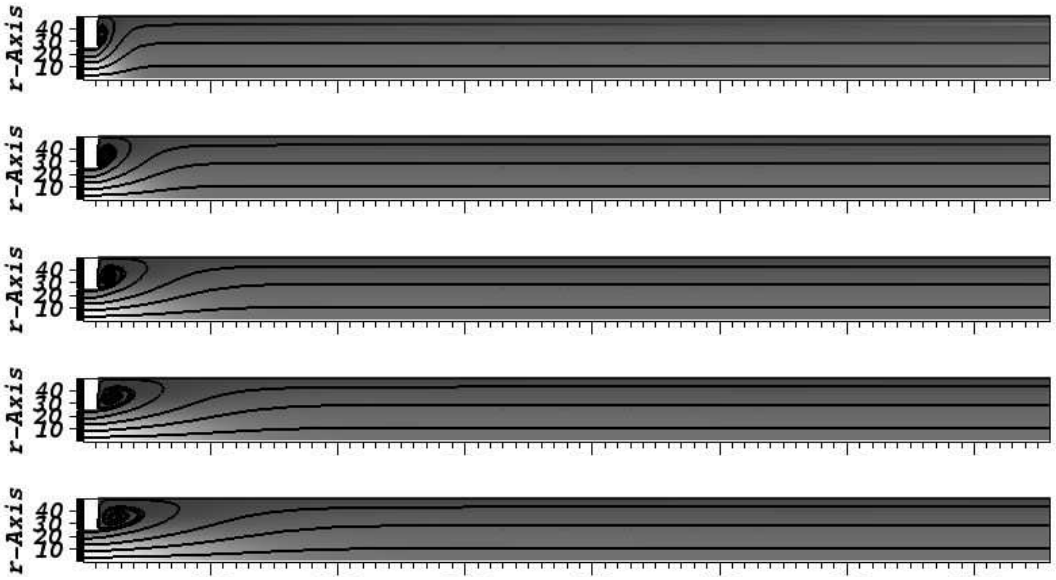

Figure 10: Flow in the x-direction with streamlines, calculated using Ilupack, continuation with step size of 50 Re, and a L_2 pipe of length 750 and grid (21, 1501; 21, 21), from top to bottom: Re 200, 400, 600, 800, 1000

The number of Newton iterations, for no continuation, eventually increases with increasing Reynolds number. This is due to an increase in nonlinearity of the system. The number of Newton iterations will increase with the curvature (nonlinearity) of the system. Newton damping will then be necessary for solving the system. While increasing the Reynolds number, eventually the initial guess of $\mathbf{0}$ will be too far off, and the equations cannot be solved in one calculation anymore. In those cases the solution to a previous calculation with a lower Reynolds number can be used (called continuation). (See for an example of the restarted calculation Table 3, the entries with Reynolds number 100, restarted from Reynolds number 50.)

The Newton method (without damping) shows quadratic convergence, as was also expected.

The time for the Newton solver and right hand side construction is similar for Ilupack

and MUMPS, these times are not shown in the tables. The number of Newton iterations at the standard Ilupack residual tolerance (1e-12), and drop tolerance (0.01) seems to be equal as well.

The number of Ilupack iteration drops during the calculation (at consecutive Newton iterations).

The number of Ilupack (GMRES) iterations, the factorisation time and the solution time remain nearly constant when increasing the Reynolds number. For the restarted calculations, the factorisation time increases. The number of Newton iterations increases for increasing Newton tolerance: more calculations are needed to achieve the required accuracy.

The solver time for higher Reynolds number (with continuation) is (almost) constant for MUMPS (the structure of the matrix does not change), whereas it increases for Ilupack, probably due to an increasing matrix condition number.

The abbreviation max. dev. in the tables indicates the maximum deviation. Average is abbreviated as avg. No. Ilupack it. indicates number of Ilupack iterations.

The number of Ilupack iterations seems to increase almost exponentially with the refinement factor of the grid, see also Figure 11.

Calculations for the nongraded grid seemed to be more stable than for the graded grid, see for example Tables 5 and 7. The calculations for the (21, 101; 21, 21) nongraded grid converged for Re=100 and Re=150, whereas they did not converge for same Reynolds numbers in the (21, 101; 21, 21) graded grid. If it works, the graded grid gives a more accurate solution than the nongraded grid. The solution for the graded (11, 51; 11, 11) grid was clearly better than for the nongraded (11, 51; 11, 11) grid.

Table 3: 2D Ilupack calculations, (11,51;11,11) graded grid, nc: no convergence

Reynolds number	Restarted (y/n)	**No. of Newton it.**	**Avg. factorisation time (s)**	Max. dev. factorisation time (s)	**Avg. solver time (s)**	Max. dev. solver time (s)	Avg. solution time (s)	Max. dev. solution time (s)	**Avg. no. of Ilupack it.**	Max. dev. no. Ilupack it.	Norm of the final residual	Newton tolerance
50	n	**6**	**0.459**	0.057	**0.050**	0.006	0.510	0.052	**12.833**	2.167	3.35E-15	1.0E-09
75	n	**6**	**0.483**	0.077	**0.051**	0.006	0.535	0.072	**12.333**	1.667	5.85E-15	1.0E-09
100	n	**8**	**0.512**	0.125	**0.053**	0.003	0.566	0.128	**12.625**	1.375	5.35E-13	1.0E-09
150	n	**nc**										1.0E-09
100	y, Re=50	**4**	**0.569**	0.051	**0.060**	0.003	0.629	0.053	**12.000**	0.000	1.03E-13	1.0E-09
50	n	**5**	**0.472**	0.056	**0.060**	0.006	0.533	0.054	**13.000**	2.000	7.59E-09	1.0E-07
75	n	**5**	**0.489**	0.077	**0.058**	0.003	0.547	0.074	**12.600**	1.400	3.35E-08	1.0E-07
100	n	**8**	**0.518**	0.114	**0.057**	0.008	0.576	0.114	**12.625**	1.375	5.35E-13	1.0E-07
150	n	**nc**										1.0E-07
100	y, Re=50	**4**	**0.552**	0.042	**0.052**	0.001	0.605	0.042	**12.000**	0.000	1.03E-13	1.0E-07
50	n	**6**	**0.457**	0.058	**0.050**	0.004	0.507	0.054	**12.833**	2.167	3.35E-15	1.0E-11
75	n	**6**	**0.480**	0.088	**0.053**	0.005	0.534	0.090	**12.333**	1.667	5.85E-15	1.0E-11
100	n	**8**	**0.517**	0.119	**0.053**	0.001	0.571	0.119	**12.625**	1.375	5.35E-13	1.0E-11
150	n	**nc**										1.0E-11
100	y, Re=50	**4**	**0.558**	0.052	**0.055**	0.001	0.614	0.052	**12.000**	0.000	1.03E-13	1.0E-11

Table 4: 2D MUMPS calculations, (11,51;11,11) graded grid, nc: no convergence

Reynolds number	Restarted (y/n)	**Number of Newton iterations**	**Average solution time (s)**	maximum deviation solution time (s)	norm of the final residual	Newton tolerance
50	n	**6**	**0.106**	0.018	3.39E-15	1.0E-09
75	n	**6**	**0.129**	0.033	5.85E-15	1.0E-09
100	n	**8**	**0.098**	0.003	5.35E-13	1.0E-09
150	n	**nc**				1.0E-09
100	y, Re=50	**4**	**0.097**	0.003	1.03E-13	1.0E-09
50	n	**5**	**0.096**	0.002	7.59E-09	1.0E-07
75	n	**5**	**0.097**	0.002	3.35E-08	1.0E-07
100	n	**8**	**0.098**	0.004	5.35E-13	1.0E-07
150	n	**nc**				1.0E-07
100	y, Re=50	**4**	**0.097**	0.002	1.03E-13	1.0E-07
50	n	**6**	**0.095**	0.002	3.39E-15	1.0E-11
75	n	**6**	**0.096**	0.003	5.85E-15	1.0E-11
100	n	**8**	**0.109**	0.003	5.35E-13	1.0E-11
150	n	**nc**				1.0E-11
100	y, Re=50	**4**	**0.109**	0.002	1.03E-13	1.0E-11

Table 5: 2D Ilupack calculations, graded grid, nc: no convergence, Newton tolerance 1e-9

Reynolds number	Restarted (y/n)	**No. of Newton it.**	**Avg. factorisation time (s)**	Max. dev. factorisation time (s)	**Avg. solver time (s)**	Max. dev. solver time (s)	Avg. solution time (s)	Max. dev. solution time (s)	**Avg. no. of Ilupack it.**	Max. dev. no. Ilupack it.	Grid
50	n	**6**	**2.382**	0.073	**0.370**	0.014	2.754	0.075	**20.667**	1.333	(21,101;21,21)
75	n	**7**	**2.474**	0.131	**0.380**	0.025	2.857	0.153	**20.429**	1.571	(21,101;21,21)
100	n	**nc**									(21,101;21,21)
150	n	**nc**									(21,101;21,21)
100	y, Re=50	**4**	**2.798**	0.161	**0.379**	0.016	3.180	0.177	**18.500**	0.500	(21,101;21,21)
50	n	**6**	**59.841**	4.940	**22.272**	4.728	82.146	9.669	**140.000**	17.000	(51,251;51,51)
75	n	**6**	**62.584**	6.099	**27.629**	5.108	86.545	8.603	**143.500**	19.500	(51,251;51,51)
100	n	**6**	**66.761**	6.099	**27.629**	5.108	94.425	10.862	**156.333**	24.667	(51,251;51,51)
150	n	**6**	**66.932**	11.381	**25.450**	8.423	92.417	19.809	**154.000**	33.000	(51,251;51,51)
100	y, Re=50	**4**	**64.689**	7.127	**28.262**	3.827	92.986	10.955	**179.500**	4.500	(51,251;51,51)

Table 6: 2D MUMPS calculations, graded grid, nc: no convergence, Newton tolerance 1e-9

Reynolds number	Restarted (y/n)	**Number of Newton iterations**	**Average solution time (s)**	maximum deviation solution time (s)	Grid
50	n	**6**	**0.561**	0.005	(21,101;21,21)
75	n	**7**	**0.561**	0.005	(21,101;21,21)
100	n	**nc**			(21,101;21,21)
150	n	**nc**			(21,101;21,21)
100	y, Re=50	**4**	**0.601**	0.005	(21,101;21,21)
50	n	**6**	**5.845**	0.205	(51,251;51,51)
75	n	**6**	**6.024**	0.078	(51,251;51,51)
100	n	**6**	**5.954**	0.415	(51,251;51,51)
150	n	**6**	**5.578**	0.380	(51,251;51,51)
100	y, Re=50	**4**	**5.818**	0.240	(51,251;51,51)

Table 7: 2D Ilupack calculations, nongraded grid, nc: no convergence, Newton tolerance 1e-9

Reynolds number	Restarted (y/n)	**No. of Newton it.**	**Avg. factorisation time (s)**	Max. dev. factorisation time (s)	**Avg. solver time (s)**	Max. dev. solver time (s)	Avg. solution time (s)	Max. dev. solution time (s)	**Avg. no. of Ilupack it.**	Max. dev. no. Ilupack it.	Grid
50	n	**6**	**0.396**	0.070	**0.051**	0.007	0.448	0.077	**12.667**	0.667	(11,51;11,11)
75	n	**6**	**0.421**	0.087	**0.049**	0.002	0.471	0.086	**12.500**	1.500	(11,51;11,11)
100	n	**8**	**0.460**	0.122	**0.050**	0.005	0.511	0.124	**12.500**	1.500	(11,51;11,11)
150	n	**nc**									(11,51;11,11)
100	y, Re=50	**4**	**0.503**	0.026	**0.049**	0.004	0.553	0.027	**11.750**	0.750	(11,51;11,11)
50	n	**6**	**2.288**	0.073	**0.412**	0.024	2.702	0.069	**20.500**	0.500	(21,101;21,21)
75	n	**6**	**2.361**	0.199	**0.415**	0.040	2.779	0.237	**20.000**	1.000	(21,101;21,21)
100	n	**6**	**2.441**	0.204	**0.390**	0.026	2.834	0.229	**20.000**	0.000	(21,101;21,21)
150	n	**8**	**2.624**	0.495	**0.395**	0.068	3.021	0.555	**19.000**	1.000	(21,101;21,21)
100	y, Re=50	**4**	**2.555**	0.175	**0.394**	0.027	2.952	0.202	**19.500**	0.500	(21,101;21,21)
50	n	**6**	**54.754**	6.761	**19.975**	6.761	74.761	10.202	**126.000**	10.000	(51,251;51,51)
75	n	**6**	**64.771**	3.131	**25.900**	1.946	90.705	1.801	**134.833**	11.833	(51,251;51,51)
100	n	**6**	**58.318**	8.252	**21.552**	4.307	79.901	10.406	**139.333**	15.333	(51,251;51,51)
150	n	**6**	**62.153**	12.995	**20.086**	3.011	82.275	15.856	**135.333**	17.667	(51,251;51,51)
100	y, Re=50	**4**	**61.067**	7.721	**23.183**	6.989	84.285	14.710	**142.000**	11.000	(51,251;51,51)

Table 8: 2D MUMPS calculations, nongraded grid, nc: no convergence, Newton tolerance 1e-9

Reynolds number	Restarted (y/n)	**Number of Newton iterations**	Average solution time (s)	maximum deviation solution time (s)	Grid
50	n	**6**	**0.097**	0.002	(11,51;11,11)
75	n	**6**	**0.098**	0.002	(11,51;11,11)
100	n	**8**	**0.096**	0.002	(11,51;11,11)
150	n	**nc**			(11,51;11,11)
100	y, Re=50	**4**	**0.096**	0.004	(11,51;11,11)
50	n	**6**	**0.556**	0.004	(21,101;21,21)
75	n	**6**	**0.555**	0.005	(21,101;21,21)
100	n	**6**	**0.556**	0.010	(21,101;21,21)
150	n	**8**	**0.554**	0.005	(21,101;21,21)
100	y, Re=50	**4**	**0.556**	0.010	(21,101;21,21)
50	n	**6**	**6.233**	0.325	(51,251;51,51)
75	n	**6**	**6.053**	0.007	(51,251;51,51)
100	n	**6**	**5.744**	0.045	(51,251;51,51)
150	n	**6**	**5.728**	0.013	(51,251;51,51)
100	y, Re=50	**4**	**5.743**	0.014	(51,251;51,51)

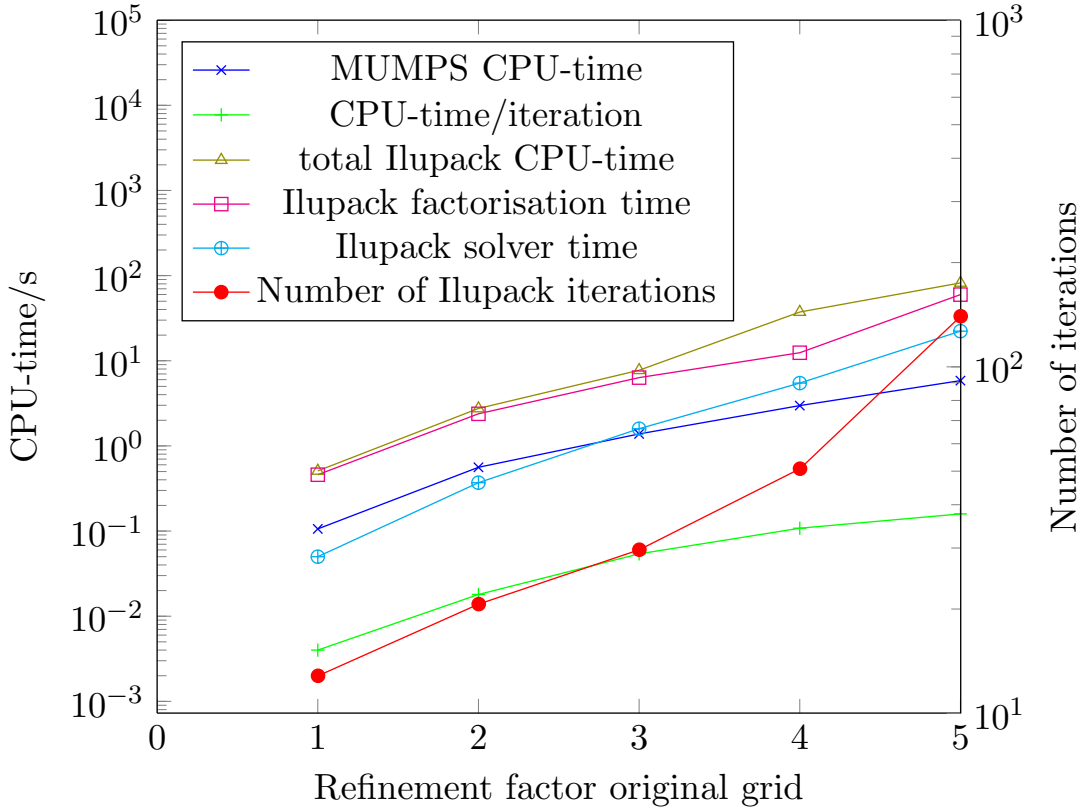

Figure 11: Re=50 graded grid, newton tol 1e-9, refinement factor compared to original (11, 51; 11, 11) grid, on the left axis the time, on the right axis the number of iterations; both vertical axis are on logarithmic scale.

4.1.1 Continuation

The continuation calculations were performed using a Reynolds number step size of 50. The pipe length was set to 750, and the grid to (21, 1501; 21, 21). The maximum number of Ilupack iterations was set to 500, this was too low for some of calculations. At lower Reynolds number this maximum number was reached. The number of Ilupack iterations was omitted from the graded grid table, the number of iterations varied from 261 to over 500. In general the number of Newton iterations for the nongraded grid seemed to be lower than for the graded grid. Choosing a higher value of the maximum number of Ilupack iterations (maxit) would have been better.

For some reason, for the graded grid at Re=100, a segmentation fault occurred, for which it did not help to restart the calculation. The calculation for Re=150 therefore used the solution from Re=50. A floating divide by 0 error occurred at Re=700 and Re=750. The calculation for Re=800 therefore used the solution from Re=650.

The calculation times etc. are very similar for the MUMPS graded and nongraded grid, therefore only those for the nongraded grid are shown.

The condition number, κ, is the ratio of the largest eigenvalue of the matrix A and its smallest eigenvalue. For the continuous Galerkin method, the convergence depends on the ratio $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, a lower κ means faster convergence (lower ratio). The Ilupack factorisa-

tion time increases for increasing Reynolds number presumably due to the fact that the condition number of the matrix increases, although this was not actually checked. For MUMPS the solution time remained constant, probably due to the fact that the structure of the matrix does not change.

The number of Newton iterations decreases with increasing Reynolds number, probably because the initial guess for each calculation gradually improves.

The ratio of Ilupack factorisation to solver time has changed dramatically, compared to the results in the previous section, from a ratio larger than one to a ratio smaller than one for the continuation. This is probably caused by the changed grid size and pipe length, since this is already valid for the first calculation (Re=50, without continuation).

4.1.2 Ilupack options

Ilupack has an option to use maximum weight matching to improve diagonal dominance of the matrix [8]. Calculations performed using the Ilupack version for Matlab on one of the problem matrices showed no real calculation time advantage of matching. A later test in the AptoFEM software failed due to a segmentation fault.

Ilupack offers several different reordering algorithms. Metisn, metise, rcm and indset gave an error when implemented in AptoFEM. The previous calculations in Matlab showed hardly any influence on calculation time. The default reordering algorithm is amd (approximate minimum degree). The other reordering algorithms tested are listed in Table 12: mmd (minimum degree), amf (approximate minimum fill), and pq (ddPQ strategy from ARMS) [8]. The ordering did not influence the number of Newton iterations or the Newton solver time. It did lead to an increase in total solution time. Therefore the default ordering amd was used.

Another Ilupack option is the drop tolerance. Entries smaller than the drop tolerance will be dropped during factorisation [8]. A higher drop tolerance may result in faster calculation, but less accurate results. The default value is 0.01. The default Schur complement drop tolerance was equal to $0.1 \times$ drop tolerance. Decreasing the drop tolerance to 0.001 lead to a decrease in solver time, but to a much more significant increase in factorisation time. Fine-tuning this ratio might still be a good idea. For a drop tolerance of 0.1, the maximum number of Ilupack iterations was reached for the grids (21, 101; 21, 21) and (51, 251; 51, 51).

Ilupack performs iterations until the error is below the residual tolerance. Its default value is 1e-12 for double precision. Raising the residual tolerance to 1e-9 slightly speeds up the factorisation and the solver time. Increasing the residual tolerance led to a decrease in the number of Ilupack iterations, as is to be expected. The double precision was used for calculation, and therefore the residual tolerance was kept at 1e-12. Furthermore lowering the residual tolerance to 1e-9 might lead to problems in the newton solver (with a tolerance of 1e-9).

The condest is the parameter describing the norm of the inverse triangular factors of the ILU decomposition [8]. A higher value can either save memory and calculation time or increase the memory and calculation time, depending on the problem. In this case increasing the condest from its default value 5 to 15 decreased the solver time, but at the same time significantly increased the factorisation time. More reordering levels were probably necessary [8]. The number of Ilupack iterations was also increased.

GMRES is by default restarted after 30 steps. This number can be changed by alter-

Table 9: 2D continuation calculations Ilupack, (21,1501;21,21) graded grid

	Reynolds number	Restarted (y,n)	**Number of Newton iterations**	linear FE solver time (s)	**Factorisation time (s)**	**Solver time (s)**	Construction time rhs (s)
avg.	50	n	**6**	127.682	**39.372**	**177.370**	30.990
max. dev				0.419	1.812	16.806	0.120
	100	y, Re=50	**segmentation fault**				
avg.	150	y, Re=50	**4**	122.648	**43.950**	**160.180**	29.867
max. dev.				0.391	0.919	7.002	0.318
avg.	200	y, Re=150	**3**	124.189	**47.479**	**165.926**	30.380
max. dev.				0.282	0.587	1.811	0.713
avg.	250	y, Re=200	**3**	124.585	**52.815**	**174.394**	30.356
max. dev.				0.736	0.740	0.818	0.302
avg.	300	y, Re=250	**3**	123.947	**57.277**	**175.081**	30.308
max. dev.				0.298	2.948	15.703	0.394
avg.	350	y, Re=300	**3**	122.724	**62.108**	**170.861**	29.970
max. dev.				0.496	3.780	19.779	0.336
avg.	400	y, Re=350	**3**	123.819	**69.033**	**170.737**	30.229
max. dev.				0.163	4.702	21.853	0.508
avg.	450	y, Re=400	**3**	124.286	**78.273**	**190.580**	30.140
max. dev.				1.231	0.230	20.900	0.199
avg.	500	y, Re=450	**3**	124.501	**84.499**	**146.935**	30.153
max. dev.				0.703	1.470	56.062	0.090
avg.	550	y, Re=500	**3**	124.332	**85.526**	**187.034**	30.219
max. dev.				0.486	6.827	43.508	0.476
avg.	600	y, Re=550	**3**	124.199	**94.364**	**183.672**	30.223
max. dev.				0.690	7.462	30.263	0.382
avg.	650	y, Re=600	**3**	124.694	**110.710**	**223.922**	30.234
max. dev.				2.166	6.618	50.561	1.077
	700	y, Re=650	**floating divide by 0**				
	750	y, Re=650	**floating divide by 0**				
avg.	800	y, Re=650	**3**	127.634	**135.797**	**221.277**	31.081
max. dev.				0.699	5.009	35.864	0.326
avg.	850	y, Re=800	**3**	132.890	**155.652**	**256.717**	31.528
max. dev.				4.745	6.992	37.464	0.890
avg.	900	y, Re=850	**3**	129.091	**164.964**	**254.299**	31.126
max. dev.				3.195	10.485	91.087	0.320
	950	y, Re=900	**1**				30.947
avg.	1000	y, Re=950	**2**	130.8736	**189.312**	**330.354**	31.258
max. dev.				1.969	9.843	23.889	0.230

Table 10: 2D continuation calculations Ilupack, (21,1501;21,21) nongraded grid

	Reynolds number	Restarted (y,n)	**Number of Newton iterations**	linear FE solver time (s)	**Factorisation time (s)**	**Solver time (s)**	**Number of Ilupack iterations**	Construction time rhs (s)
avg.	50	n	**6**	132.289	**37.541**	**217.961**		31.418
max. dev.				0.870	1.878	40.424		0.219
avg.	100	y, Re=50	**4**	130.275	**41.954**	**203.057**		31.262
max. dev.				2.961	2.391	61.986		0.664
avg.	150	y, Re=100	**4**	128.986	**40.984**	**162.115**	**391.750**	31.003
max. dev.				4.898	4.677	83.472	90.250	0.865
avg.	200	y, Re=150	**3**	129.955	**47.368**	**158.668**	**403.000**	31.073
max. dev.				1.464	2.073	36.797	124.000	0.367
avg.	250	y, Re=200	**3**	126.222	**53.066**	**134.963**		30.732
max. dev.				1.798	1.315	47.029		0.499
avg.	300	y, Re=250	**3**	122.372	**53.892**	**113.869**	**320.333**	29.627
max. dev.				0.383	3.450	50.661	130.667	0.176
avg.	350	y, Re=300	**3**	128.191	**64.563**	**152.497**	**349.333**	30.698
max. dev.				0.130	4.969	7.263	39.333	1.112
avg.	400	y, Re=350	**3**	125.511	**66.273**	**135.330**	**340.000**	30.570
max. dev.				2.533	3.985	11.097	30.000	0.781
avg.	450	y, Re=400	**3**	125.330	**74.361**	**111.620**	**275.667**	30.602
max. dev.				3.213	1.510	28.091	65.333	0.807
avg.	500	y, Re=450	**3**	128.405	**83.807**	**172.964**	**275.667**	30.888
max. dev.				4.894	2.627	75.340	65.333	1.197
avg.	550	y, Re=500	**3**	124.126	**87.548**	**173.883**	**297.000**	30.526
max. dev.				1.026	7.103	40.473	18.000	0.659
avg.	600	y, Re=550	**3**	126.661	**91.623**	**160.478**	**337.667**	30.826
max. dev.				1.739	0.500	10.732	3.333	0.296
avg.	650	y, Re=600	**3**	129.467	**96.531**	**175.662**	**343.667**	31.186
max. dev.				2.297	10.443	42.461	71.667	0.316
avg.	700	y, Re=650	**3**	125.594	**97.511**	**131.551**	**286.000**	30.523
max. dev.				3.101	6.997	33.206	86.000	0.977
avg.	750	y, Re=700	**3**	125.754	**103.716**	**169.859**	**356.667**	30.915
max. dev.				2.158	7.380	26.597	61.667	0.241
avg.	800	y, Re=750	**3**	126.187	**120.291**	**249.229**	**382.000**	31.274
max. dev.				2.254	7.338	28.797	20.000	0.415
avg.	850	y, Re=800	**3**	129.642	**122.219**	**210.005**	**335.000**	31.190
max. dev.				2.667	3.443	99.186	42.000	0.392
avg.	900	y, Re=850	**3**	126.163	**125.066**	**242.845**	**409.333**	30.872
max. dev.				2.435	14.167	37.047	24.667	0.950
avg.	950	y, Re=900	**3**	128.238	**131.516**	**176.950**	**336.667**	31.029
max. dev.				2.233	9.668	40.174	62.667	0.356
avg.	1000	y, Re=950	**2**	127.653	**131.832**	**174.028**	**320.500**	30.985
max. dev.				0.320	7.160	26.871	41.500	0.089

Table 11: 2D continuation calculations MUMPS, (21,1501;21,21) nongraded grid

	Reynolds number	Restarted (y,n)	**Number of Newton iterations**	linear FE solver time (s)	**Solution time (s)**	Construction time rhs (s)
avg.	50	n	**6**	127.274	**8.357**	32.368
max. dev.				1.748	0.143	0.322
avg.	100	y, Re=50	**4**	127.196	**8.427**	32.545
max. dev.				1.670	0.266	0.356
avg.	150	y, Re=100	**3**	126.751	**8.284**	32.561
max. dev.				0.378	0.061	0.281
avg.	200	y, Re=150	**3**	126.988	**8.486**	32.374
max. dev.				0.318	0.173	0.140
avg.	250	y, Re=200	**3**	127.558	**8.507**	32.604
max. dev.				0.302	0.009	0.342
avg.	300	y, Re=250	**3**	126.029	**8.389**	32.122
max. dev.				0.540	0.041	0.186
avg.	350	y, Re=300	**3**	128.031	**8.584**	32.724
max. dev.				0.797	0.111	0.458
avg.	400	y, Re=350	**3**	126.628	**8.480**	32.359
max. dev.				1.034	0.253	0.463
avg.	450	y, Re=400	**3**	125.900	**8.405**	32.233
max. dev				1.092	0.236	0.360
avg.	500	y, Re=450	**3**	127.648	**8.369**	32.580
max. dev.				1.003	0.020	0.363
avg.	550	y, Re=500	**3**	125.830	**8.335**	32.150
max. dev.				0.849	0.235	0.176
avg.	600	y, Re=550	**3**	125.993	**8.330**	32.429
max. dev.				0.494	0.079	0.304
avg.	650	y, Re=600	**3**	125.489	**8.338**	32.304
max. dev.				0.509	0.061	0.198
avg.	700	y, Re=650	**3**	126.794	**8.396**	32.597
max. dev.				1.820	0.226	0.600
avg.	750	y, Re=700	**3**	128.065	**8.403**	32.810
max. dev.				0.067	0.108	0.315
avg.	800	y, Re=750	**3**	126.710	**8.746**	32.564
max. dev.				0.709	0.374	0.401
avg.	850	y, Re=800	**3**	125.603	**8.256**	32.197
max. dev.				0.658	0.009	0.030
avg.	900	y, Re=850	**3**	126.937	**8.547**	32.607
max. dev.				1.935	0.283	0.348
avg.	950	y, Re=900	**3**	126.565	**8.465**	32.502
max. dev.				0.924	0.176	0.426
avg.	1000	y, Re=950	**2**	127.928	**8.732**	33.003
max. dev.				0.308	0.309	0.358

Table 12: Different ordering algorithms for Ilupack for the (51, 251; 51, 51) mesh, 2D calculations

Reynolds number	**Ordering**	Avg. factorisation time (s)	Max. dev. factorisation time (s)	Avg. solver time (s)	Max. dev. solver time (s)	**Avg. solution time (s)**	Max. dev. solution time (s)	Avg. no. of Ilupack it.	Max. dev. no. Ilupack it.
50	**amd**	19.456	0.843	9.457	1.596	**28.929**	1.454	65.000	10.000
100	**amd**	21.273	0.957	11.047	1.483	**32.338**	2.325	60.667	1.667
50	**mmd**	36.497	0.903	8.801	0.668	**45.313**	1.489	63.667	2.333
100	**mmd**	40.281	3.273	10.097	2.012	**50.394**	5.284	60.833	1.167
50	**amf**	21.857	2.080	11.437	2.342	**33.310**	4.423	61.167	1.167
100	**amf**	19.485	1.349	8.359	2.149	**27.860**	3.497	56.167	3.167
50	**pq**	62.715	5.088	23.453	5.135	**86.199**	8.824	140.000	17.000
100	**pq**	63.131	10.433	25.707	8.155	**88.873**	18.595	156.333	24.667

Table 13: Ilupack options for the (51, 251; 51, 51) grid, 2D calculations

Reynolds number	**Non-standard options**	Avg. factorisation time (s)	Max. dev. factorisation time (s)	Avg. solver time (s)	Max. dev. solver time (s)	**Avg. solution time (s)**	Max. dev. solution time (s)	Avg. no. of Ilupack it.	Max. dev. no. Ilupack it.
50		19.456	0.843	9.457	1.596	**28.929**	1.454	65.000	10.000
100		21.273	0.957	11.047	1.483	**32.338**	2.325	60.667	1.667
50	**restol=1e-9**	19.119	0.449	7.590	0.574	**26.723**	0.347	52.833	2.833
100	**restol=1e-9**	19.671	0.602	7.152	0.730	**26.838**	0.930	49.833	2.167
50	**droptol=0.001**	51.789	5.216	3.808	0.166	**55.614**	5.050	16.833	0.833
100	**droptol=0.001**	59.826	7.485	4.308	0.687	**64.150**	8.153	16.333	0.667
50	**condest=15**	55.952	8.999	4.183	0.650	**60.152**	9.381	16.833	0.833
100	**condest=15**	63.243	2.987	4.628	0.348	**67.889**	2.640	16.333	0.667
50	**elbow=20**	20.822	1.924	11.609	2.974	**32.447**	4.064	65.000	10.000
100	**elbow=20**	21.361	1.329	10.592	2.198	**31.969**	3.095	60.667	1.667
50	**nrestart=20**	19.165	0.622	14.725	0.388	**33.906**	0.485	102.333	3.333
100	**nrestart=20**	19.997	0.341	13.362	1.256	**33.374**	1.265	92.500	8.500
50	**nrestart=40**	19.317	0.684	7.677	1.735	**27.010**	1.745	50.167	9.167
100	**nrestart=40**	20.175	0.451	6.596	0.619	**26.787**	1.071	41.000	0.000

ing the nrestart option in Ilupack. A smaller value reduces memory and a larger value can improve convergence [8]. This is also visible in the results: increasing the nrestart decreases the number of Ilupack iterations necessary and the solution time and vice versa. Decreasing the memory use did not lead to a large speed up in the results in this case.

4.2 MUMPS vs Ilupack (spectral) 3D calculations

In this section the spectral method combined with FEM Ilupack is compared to the spectral method combined with FEM MUMPS. Most time appeared to be spent in assembling the matrix, therefore the calculations were partially performed in parallel: the matrix assembly was performed in parallel, not the actual solving. Using 4 processors speeds this up by a factor of around 3-4 (around 1800 seconds versus around 460 seconds for Ilupack, and 3150 seconds versus 850 for MUMPS).

The calculations were performed on ironman, a server which is a HP DL380p Gen8 with 2 x Intel Xeon E5-2690 with 8-core processors (32 logical cores total) and 256 GB RAM. The calculations were performed for the (21, 101; 21, 21) graded grid. The executables were compiled with the 03 flag, which also reduced the calculation time considerably.

The 3D Ilupack calculations sometimes gave a memory allocation error, probably caused by more intense server occupation. The same calculation at another moment did give the right results.

For some reason the time for the linear FE solver was lower for Ilupack than for MUMPS. The compilation flags were checked, but were equal. It was expected these times would be similar, like in the 2D calculations. The same number of processors for the parallel matrix assembly were used, therefore this should not be the problem either. This difference in linear FE solver time lead to the fact that the total calculation times for MUMPS and Ilupack were similar.

The 3D calculations were also performed for a slightly perturbed, nonsymmetric inflow condition (*i.e.* the inflow depends on the value of θ). These calculations were performed as an initial test, to find out whether this was possible. Since it proved to be possible, the theory of perturbations causing the asymmetry at $Re = 1149 \pm 10$ can be tested, see also subsection 1.5. (However due computation time, this was not possible in this research.) The function used to describe this θ-dependent inflow condition was:

$$u_i(0, r, \theta_i) = (1 - r^2) + 0.1(1 - r^2)\sin(\frac{2\pi(i-1)}{N}). \tag{46}$$

Here i is the number of the collocation point it is at, and N is the total number of collocation points.

The calculation times and the number of iterations (both Newton, and for Ilupack the Ilupack iterations) for symmetric and perturbed inlet velocity were very similar, therefore the tables with the perturbed inlet velocity were omitted.

The calculation of the right hand side for the first Newton iteration sometimes seemed to take much longer than for the following Newton iterations. For MUMPS this effect seems larger than for Ilupack. This might be caused by the initial guess being far off for the first Newton iteration. This effect was not observed for the 2D calculations.

The factorisation time in the 3D calculation seems to be relatively larger than for the 2D calculation, therefore decreasing the drop tolerance, and fine tuning might be worth the effort.

The 3D calculations with perturbed inlet velocity appeared to work, see also Figure 12. However, the interesting Reynolds numbers (of around 1140) could not be reached within the time span of this research project.

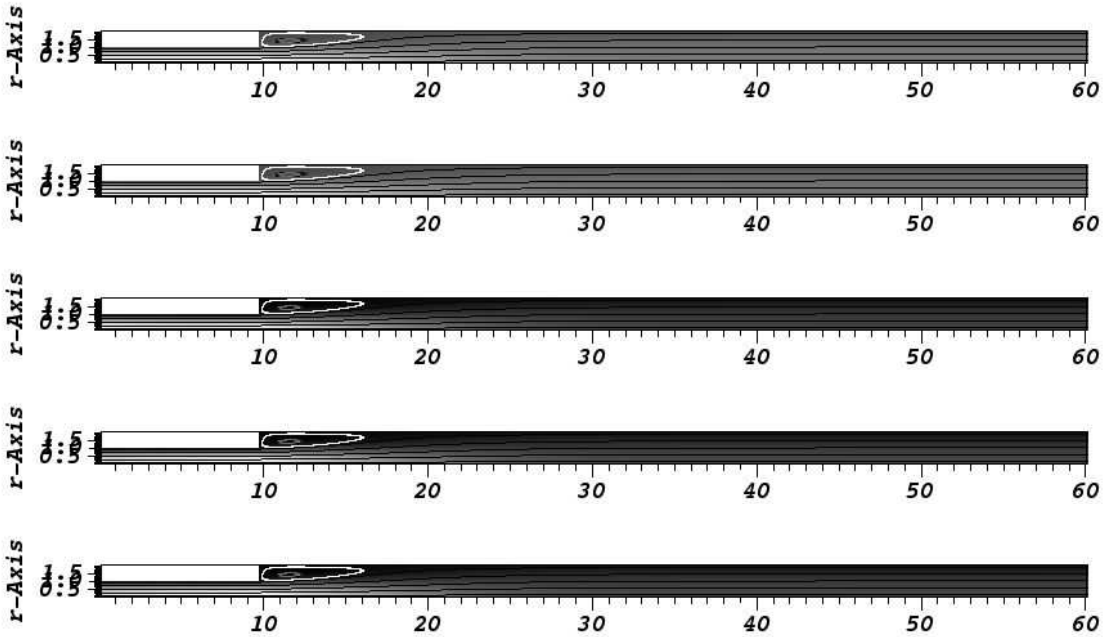

Figure 12: Flow in the x-direction with streamlines, calculated using Ilupack, continuation with step size of 10 Re, and a L_2 pipe of length 50 at Re=100, for 10 collocation points and (21, 101; 21, 21) grid, from top to bottom collocation point number: 1, 3, 5, 7, 9

Table 14: 3D Ilupack spectral continuation calculations, (21,101;21,21) grid, symmetric inlet

Reynolds number	Restarted (y/n)	**Number of Newton iterations**	Newton iteration number	**linear FE solver time (s)**	**Factorisation time (s)**	**Solver time (s)**	**Number of Ilupack iterations**	Total Ilupack solution time (s)	**Construction time rhs (s)**	Norm of the Residual	Damping factor	Order of convergence
10	n	4	0						**7.988**	2.74E-01	1	
			1	479.575	304.424	221.166	145	527.177	2.343	1.76E-01	1	
			2	471.304	300.202	235.735	149	537.492	2.347	1.11E-02	1	6.31E+00
			3	464.453	296.218	218.491	142	516.260	2.369	1.13E-05	1	2.49E+00
			4	477.916	303.071	229.137	147	533.787	2.357	3.45E-11	1	1.84E+00
average				**473.312**	**300.979**	**226.132**	**145.750**	528.679	3.480			
max. dev.				8.859	4.761	9.603	3.750	12.419	4.507			
20	y, Re=10	4	0						**7.983**	5.98E-01	1	
			1	496.398	306.752	256.908	176	565.174	2.361	4.07E-03	1	
			2	482.235	308.313	206.995	141	516.814	2.360	3.64E-05	1	9.45E-01
			3	483.818	303.661	207.199	141	512.375	2.390	1.27E-09	1	2.18E+00
			4	502.216	311.872	222.825	146	536.209	2.366	2.97E-14	1	1.04E+00
average				**491.167**	**307.649**	**23.482**	**151.000**	532.643	3.492			
max. dev.				11.049	4.222	33.426	25.000	32.531	4.491			
30	y, Re=20	3	0						3.130	1.99E-01	1	
			1	476.126	315.659	221.694	148	538.877	2.361	2.10E-03	1	
			2	476.116	314.665	206.537	138	522.728	2.356	1.63E-05	1	1.07E+00
			3	492.464	347.297	270.957	145	619.881	2.356	4.58E-10	1	2.16E+00
average				**481.568**	**325.874**	**233.063**	**143.667**	560.495	2.550			
max.	dev.			10.896	21.423	37.894	5.667	59.386	0.579			
40	y, Re=30	3	0						**4.039**	9.96E-02	1	
			1	475.271	317.174	259.164	175	577.860	2.353	1.12E-03	1	

			2	477.915	316.129	206.310	139	523.962	2.363	5.93E-06	1	1.17E+00
			3	488.922	312.792	229.743	148	544.066	2.393	7.95E-11	1	2.14E+00
average				**480.703**	**315.365**	**231.739**	**154.000**	548.630	2.787			
max.	dev.			8.219	2.573	27.425	21.000	29.231	1.253			
50	y, Re=40	3	0						2.372	5.99E-02	1	
			1	516.167	363.939	232.075	143	597.551	2.406	6.67E-04	1	
			2	517.299	375.849	195.022	123	572.409	2.370	2.45E-06	1	1.25E+00
			3	503.987	323.965	235.224	155	560.709	2.397	1.61E-11	1	2.13E+00
average				**512.484**	**354.584**	**220.774**	**140.333**	576.890	2.386			
max. dev.				8.497	30.619	25.751	17.333	20.662	0.020			
60	y, Re=50	3	0						**6.113**	4.00E-02	1	
			1	494.539	341.852	243.223	148	586.625	2.360	4.37E-04	1	
			2	491.701	345.303	233.453	142	580.327	2.362	1.15E-06	1	1.32E+00
			3	483.417	343.481	221.714	135	566.762	2.360	4.00E-12	1	2.12E+00
average				**489.886**	**343.545**	**232.797**	**141.667**	577.904	3.298			
max. dev.				6.469	1.757	11.082	6.667	11.143	2.815			
70	y, Re 60	3	0						2.372	2.86E-02	1	
			1	501.983	360.426	204.655	124	566.602	2.410	3.06E-04	1	
			2	491.057	337.515	279.166	146	618.312	2.351	6.01E-07	1	1.37E+00
			3	462.855	332.491	310.031	155	644.151	2.357	1.19E-12	1	2.11E+00
average				**485.298**	**343.477**	**264.617**	**141.667**	609.688	2.372			
max. dev.				22.444	16.949	59.962	17.667	43.086	0.037			
80	y, Re=70	3	0						3.857	2.15E-02	1	
			1	484.676	343.970	277.708	175	623.235	2.367	2.25E-04	1	
			2	483.174	347.017	278.745	176	627.315	2.371	3.41E-07	1	1.42E+00
			3	491.801	345.024	228.588	144	575.175	2.369	4.12E-13	1	2.10E+00
average				**486.550**	**345.337**	**261.680**	**165.000**	608.575	2.741			
max. dev.				5.251	1.680	33.092	21.000	33.400	1.117			
90	y, Re=80	3	0						3.663	1.68E-02	1	
			1	508.056	361.203	237.088	141	599.863	2.369	1.72E-04	1	
			2	491.309	374.309	234.571	143	610.452	2.358	2.07E-07	1	1.47E+00

			3	490.066	361.607	240.364	145	603.534	3.405	1.60E-13	1	2.09E+00
average				**496.477**	**365.706**	**237.341**	**143.000**	604.616	2.949			
max. dev.				11.579	8.603	3.023	2.000	5.836	0.715			
100	y, Re=90	3	0						2.381	1.35E-02	1	
			1	501.184	405.604	225.668	121	632.887	2.378	1.36E-04	1	
			2	497.252	407.513	260.548	141	669.680	2.358	1.33E-07	1	1.51E+00
			3	489.967	406.775	232.466	124	640.869	2.364	6.83E-14	1	2.09E+00
average				**496.134**	**406.631**	**239.561**	**128.667**	647.812	2.370			
max. dev.				6.168	1.027	20.988	12.333	21.868	0.012			

Table 15: 3D MUMPS spectral continuation calculations, (21,101;21,21) grid, symmetric inlet

Reynolds number	Restarted (y/n)	**Number of Newton iterations**	Newton iteration number	**linear FE solver time (s)**	**Total MUMPS solution time (s)**	**Construction time rhs (s)**	Norm of the residual	Damping factor	Order of Convergence
10	n	4	0			**347.418**	2.74E-01	1	
			1	820.107	138.345	4.715	1.76E-01	1	
			2	810.487	139.211	4.730	1.11E-02	1	6.31E+00
			3	851.070	149.441	4.880	1.13E-05	1	2.49E+00
			4	853.576	154.376	4.918	3.45E-11	1	1.84E+00
average				**833.810**	**145.343**	73.332			
max. dev.				23.323	9.032	274.086			
20	y, Re=10	4	0			**6.218**	5.98E-01	1	
			1	841.777	143.070	4.965	4.07E-03	1	
			2	865.648	152.877	5.232	3.64E-05	1	9.45E-01
			3	872.684	143.870	4.991	1.27E-09	1	2.18E+00
			4	868.332	148.854	4.960	2.99E-14	1	1.04E+00
average				**862.110**	**147.168**	5.273			
max. dev.				20.333	5.709	0.945			
30	y, Re=20	3	0			**142.177**	1.99E-01	1	
			1	1068.398	174.218	6.207	2.10E-03	1	
			3	988.835	162.063	5.599	1.63E-05	1	1.07E+00
			3	986.262	163.492	7.682	4.58E-10	1	2.16E+00
average				**1014.498**	**166.591**	40.416			
max. dev.				53.899	7.627	101.761			
40	y, Re=30	3	0			**28.957**	9.96E-02	1	
			1	970.137	155.179	5.106	1.12E-03	1	

			2	998.416	151.628	4.885	5.93E-06	1	1.17E+00
			3	973.622	166.869	5.439	7.95E-11	1	2.14E+00
average				**980.725**	**157.892**	11.097			
max. dev.				17.691	8.977	17.860			
50	y, Re=40	3	0			**13.204**	5.99E-02	1	
			1	833.319	141.501	4.792	6.67E-04	1	
			2	842.326	143.390	4.783	2.45E-06	1	1.25E+00
			3	841.478	142.725	4.796	1.61E-11	1	2.13E+00
average				**839.041**	**142.539**	6.894			
max. dev.				5.722	1.038	6.310			
60	y, Re=50	3	0			4.790	4.00E-02	1	
			1	833.987	142.372	4.765	4.37E-04	1	
			2	826.878	140.651	4.749	1.15E-06	1	1.32E+00
			3	825.542	140.563	4.769	4.00E-12	1	2.12E+00
average				**828.803**	**141.195**	4.769			
max. dev.				5.185	1.177	0.022			
70	y, Re=60	3	0			**11.769**	2.86E-02	1	
			1	841.244	138.637	4.779	3.06E-04	1	
			2	841.197	138.881	4.719	6.01E-07	1	1.37E+00
			3	811.725	139.534	4.721	1.19E-12	1	2.11E+00
average				**831.389**	**139.017**	6.497			
max. dev.				19.664	0.517	5.272			
80	y, Re=70	3	0			**10.054**	2.15E-02	1	
			1	847.609	137.395	4.820	2.25E-04	1	
			2	845.771	137.561	4.787	3.41E-07	1	1.42E+00
			3	846.269	137.412	4.841	4.12E-13	1	2.10E+00
average				**846.550**	**137.456**	6.126			
max. dev.				1.059	0.105	3.929			
90	y, Re=80	3	0			4.910	1.68E-02	1	
			1	858.219	149.642	4.926	1.72E-04	1	
			2	865.244	153.996	4.942	2.07E-07	1	1.47E+00

			3	864.796	151.177	4.925	1.60E-13	1	2.09E+00
average				**862.753**	**151.605**	4.926			
max. dev.				4.534	2.391	0.016			
100	y, Re=90	3	0			5.568	1.35E-02	1	
			1	851.857	147.413	4.850	1.36E-04	1	
			2	864.430	163.369	4.730	1.33E-07	1	1.51E+00
			3	840.774	163.373	4.768	6.83E-14	1	2.09E+00
average				**852.354**	**158.052**	4.979			
max. dev.				12.076	10.639	0.589			

5 Conclusion and discussion

In this research the flow of an incompressible fluid through a 1:2 sudden expansion of a pipe was modelled. The first model consisted of a symmetry-based reduction to a 2D calculation, and the second model used a Fourier spectral method in the azimuthal direction. MUMPS, a direct solver, was compared to Ilupack, a Krylov subspace based iterative solver. Both solvers were implemented in the AptoFEM package. The MUMPS *solver* was faster in all calculations.

For the 2D calculations, MUMPS was significantly faster, since the number of degrees of freedom in these calculations were reasonably low. In these calculations, the number of Ilupack iterations appeared to increase almost exponentially with the refinement factor of the grid. The calculations for the nongraded grid seemed to be more stable than for the graded grid.

For an increasing grid refinement factor the ratio of Ilupack factorisation time to solver time decreased. In the non-restarted (no continuation) calculations performed the Ilupack factorisation time was always higher than for the restarted calculations. The ratio of Ilupack factorisation to solver time decreased dramatically for the calculations performed in the continuation subsection 4.1.1, probably due to the increased grid size and pipe length.

The number of Newton iterations decreased with increasing Reynolds number in continuation calculations, probably because the initial guess for each calculation gradually improved, as $\frac{\Delta Re}{Re_{new}}$ became a smaller ratio. The constant change in Reynolds number was therefore expected to have a smaller influence on the flow for higher Re_{new}.

The spectral discretisation was successfully implemented in the azimuthal direction. For the 3D calculations, MUMPS itself solves faster than Ilupack, but since the linear FE solver time for MUMPS is much higher than for Ilupack, the calculation times for MUMPS and Ilupack 3D calculations were similar, and in some cases the total Ilupack calculation time was actually lower than for MUMPS. This was true for the 3D-calculations as well, though this might not be the case for larger systems. Eventually the system is expected be too large for MUMPS to solve, in which case Ilupack is expected to be better.

The 3D calculations with perturbed inlet velocity appeared to work. However, the interesting Reynolds numbers (of around 1140) could not be reached within the time span of this research project.

5.1 Suggestions for future research

Due the computational complexity, it was not possible to go up to the interesting Reynolds numbers (1100-1200, see subsection 1.5) for the spectral collocation method, with both MUMPS and Ilupack. This could be done in a future research project. This will be particularly interesting for the calculations with the small distortion in the inlet velocity. Perhaps small distortions, with a θ-dependency, in the pipe geometry might also be interesting to investigate. This implementation could prove a more difficult than for the inlet velocity distortion.

A useful improvement is to automate the continuation to higher Reynolds numbers. The step size for continuation was chosen to be constant during this research (50 for 2D calculations, 10 for 3D calculations). It might be a good idea to look into a better continuation procedure in order to be able to reach higher Reynolds numbers, such as the

pseudo arclength continuation method.

The Ilupack settings were roughly optimized for the 2D calculations without continuation. However since the ratio of factorisation time to solver time changed from higher than 1 to lower than 1 for switching from 2D no continuation to 2D continuation, it might be a better idea to optimise the results for the continuation. Furthermore, for the 3D calculations with continuation, other settings might be better. However, the nrestart and drop tolerance should be verified. The nrestart determines the number of iterations before GMRES is restarted. Since the 3D calculations require much more memory than the 2D calculations, a lower value of nrestart might decrease the memory usage, and therefore the solution time. The factorisation time in the 3D calculation seems to be relatively larger than for the 2D calculation, therefore decreasing the drop tolerance, and fine-tuning may be worthwhile.

Full 3D-FEM calculations were not possible, since the required elements, their basis functions, etc. currently are not available in AptoFEM.

All calculations with the spectral collocation method were performed using 10 collocation points, encouraging computational simplicity. After 3D-optimization, and using parallel matrix assembly, calculations using more collocation points could be possible.

5.2 Acknowledgements

I would like to express my gratitude to my supervisors for all the help they offered during my research, especially Ed in implementing Ilupack into AptoFEM. Furthermore I would like to thank the people in my office for helping me out when I had a question. Finally I would like to thank the Lifelong Learning Programme and the Groninger Universiteits Fonds for their financial support during my stay in Nottingham.

References

[1] *COMSOL Multiphysics Reference Guide.*

[2] *MUltifrontal Massively Parallel Solver (mumps 4.10.0) Users's guide*, May 2011.

[3] P. R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications **23** (2001), no. 1, 15–41.

[4] P.R. Amestoy, I.S. Duff, and J.-Y. L'Excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Computer Methods in Applied Mechanics and Engineering (2000).

[5] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet, *Hybrid scheduling for the parallel solution of linear systems*, Parallel Computing **32** (2006), no. 2, 136–156.

[6] G. Biswas, M. Breuer, and F. Durst, *Backward-facing step flows for various expansion ratios at low and moderate Reynolds numbers*, Journal of Fluids Engineering **126** (2004), 362–374.

[7] M. Bollhöfer and Y. Saad, *Multilevel preconditioners constructed from inverse-based ILUs*, SIAM Journal on Scientific Computing **27** (2006), no. 5, 1627–1650.

[8] M. Bollhöfer and Y. Saad, *Ilupack - preconditioning software package v2.4*, 06 2011.

[9] J.P. Boyd, *Chebyshev and Fourier spectral methods*, 2 ed., Dover Publications, Inc., 2000.

[10] C.D. Cantwell, D. Barkley, and H.M. Blackburn, *Transient growth analysis of flow through a sudden expansion in a circular pipe*, Physics of Fluids **22** (2010).

[11] C. Canuto, Y. Maday, and A. Quarteroni, *Analysis of the combined Finite Element and Fourier interpolation*, Numerische Mathematik **39** (1982), 205–220.

[12] C. Canuto, Y. Maday, and A. Quarteroni, *Combined Finite Element and Spectral approximation of the Navier-Stokes equations*, Numerische Mathematik **44** (1984), 201–217.

[13] K.A. Cliffe, E.J.C. Hall, and P. Houston, *Adaptive discontinuous Galerkin methods for eigenvalue problems arising in incompressible fluid flows*, SIAM Journal on Scientific Computing (2010).

[14] K.A. Cliffe, E.J.C. Hall, and P. Houston, *hp-adaptive discontinuous Galerkin methods for bifurcation phenomena in open flows*, Computers & Mathematics with Applications (2012).

[15] K.A. Cliffe, E.J.C. Hall, P. Houston, E.T. Phipps, and A.G.Salinger, *Adaptivity and a posteriori error control for bifurcation problems III: Incompressible fluid flow in open systems with O(2) symmetry*, Journal of Scientific Computing **52** (2012), 153–179.

[16] K.A. Cliffe, E.J.C. Hall, P. Houston, E.T. Phipps, and A.G. Salinger, *Adaptivity and a posteriori error control for bifurcation problems II: Incompressible fluid flow in open systems with Z_2 symmetry*, Journal of Scientific Computing **47** (2011), 389–418.

[17] N. Coleman, *A derivation of the Navier-Stokes equations*, B.S. Undergraduate Mathematics Exchange **7** (2010), no. 1, 20–26.

[18] H. Elman, D. Silvester, and A. Wathen, *Finite elements and fast iterative solvers: with applications to incompressible fluid dynamics*, 1 ed., Oxford University Press, UK, May 2005.

[19] G. Fasshauer, *Chapter 14 Arnoldi iteration and GMRES*, Class notes, 2006, Illinois Institute of Technology.

[20] R.M. Fearn, T. Mullin, and K.A. Cliffe, *Nonlinear flow phenomena in a symmetric sudden expansion*, Journal of Fluid Mechanics **211** (1999), 595–608.

[21] C.A. Felippa, *Nonlinear Finite Element Methods*, 2012.

[22] A. Févre, *Short presentation of MUMPS*, October 2006.

[23] M. Golubitsky, I. Stewart, and D.G. Schaeffer, *Singularities and Groups in Bifurcation Theory*, 1 ed., vol. 2, Springer-Verlag, 1988.

[24] M.D. Griffith, T. Leweke, M.C. Thompson, and K. Hourigan, *Effect of small asymmetries on axisymmetric stenotic flow*, Journal of Fluid Mechanics **721** (2013), no. 1, 1–11.

[25] E.J.C. Hall, *G14VMS variational methods*, Lecture notes, 2012, University of Nottingham.

[26] Wolfram Research Inc., *Sin*, website, 2013.

[27] S.P. Kiselev, E.V. Vorozhtsov, and V.M. Fomin, *Foundations of Fluid Mechanics with Applications; Problem Solving Using Mathematica*, Birkhäuser, 1999.

[28] P. Matthews and K.A. Cliffe, *G14ADE Advanced Techniques for Differential Equations*, Lecture notes, 2013, University of Nottingham.

[29] T. Mullin, J.R.T. Seddon, M.D. Mantle, and A.J. Sederman, *Bifurcation phenomena in the flow through a sudden expansion in a circular pipe*, Physics of Fluids **21** (2009).

[30] J. Núñez, E. Ramos, and J.M. Lopez, *A mixed Fourier-Galerkin-finite-volume method to solve the fluid dynamics equations in cylindrical geometries*, Fluid Dynamics Research **44** (2012), no. 3.

[31] Y. Saad, *Iterative methods for sparse linear systems*, 2 ed., Society for Industrial and Applied Mathematics, 2003.

[32] Y. Saad and M.H. Schulz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific Computing **7** (1986), no. 3, 856–869.

[33] E. Sanmiguel-Rojas and T. Mullin, *Finite-amplitude solutions in the flow through a sudden expansion in a circular pipe*, Journal of Fluid Mechanics **691** (2012), 201–213.

[34] P.R. Tulip, *Dielectric and lattice dynamical properties of molecular crystals via density functional perturbation theory: Implementation within a first principles code*, Ph.D. thesis, The University of Durham, 2004.

[35] J. Vétel, A. Garon, D. Pelletier, and M.-I. Farinas, *Asymmetry and transition to turbulence in a smooth axisymmetric constriction*, Journal of Fluid Mechanics **607** (2008), 351–386.

[36] F.W. Wubs, *Computational Methods of Science*, Lecture notes, 2012, University of Groningen.

[37] F.W. Wubs and D.A. Dijkstra, *Bifurcation analysis of fluid flows: Analysis beyond simulation*, Lecture notes, 2013, Universities of Groningen and Utrecht.

A Various definitions of Re

Cliffe *et al.* use the following definition for the Reynolds number (for example in [15]):

$$Re = \frac{u_{\max} R_1}{\nu}. \tag{47}$$

Here u_{max} is the peak inlet velocity, and R_1 is the radius of the inlet section of the pipe, and ν is the kinematic viscosity, defined by:

$$\nu = \frac{\rho}{\mu}. \tag{48}$$

On the other hand, Mullin *et al.* [29] use another definition:

$$Re = \frac{u_{\text{av}} D_1}{\nu}. \tag{49}$$

Here u_{av} is the mean value of the velocity of the incoming flow, and D_1 is the diameter of the inlet pipe. The relation between those definitions is further examined; obviously, $D_1 = 2R_1$. At the inlet pipe, a Poiseuille flow profile is often assumed. When R_1 has length 1, and using a no-slip condition, the velocity profile can be written as $u(r) = (1 - r^2)u_{\max}$. The mean velocity is equal to:

$$v_{\text{avg}} = \frac{\int_0^1 2\pi u_{\max}(1 - r^2) r \mathrm{d}r}{\pi R_1^2} = \frac{1}{2} u_{\max}. \tag{50}$$

Eventually for this case (with $R_1 = 1$), both definitions of the Reynolds number, result in the same Reynolds number.

B Cartesian to cylindrical coordinates for Navier-Stokes

Define:

$$\begin{aligned} y &= r\sin(\theta), \\ z &= r\cos(\theta). \end{aligned} \tag{51}$$

This implies:

$$\begin{aligned} u_y &= \frac{\partial y}{\partial t} = \sin(\theta)\frac{\partial r}{\partial t} + r\cos(\theta)\frac{\partial \theta}{\partial t} = \sin(\theta)u_r + \cos(\theta)u_\theta, \\ u_z &= \frac{\partial z}{\partial t} = \cos(\theta)\frac{\partial r}{\partial t} - r\sin(\theta)\frac{\partial \theta}{\partial t} = \cos(\theta)u_r - \sin(\theta)u_\theta. \end{aligned} \tag{52}$$

Here:

$$\begin{aligned} \frac{\partial y}{\partial t} &= u_y, \\ \frac{\partial z}{\partial t} &= u_z, \\ \frac{\partial r}{\partial t} &= u_r, \\ r\frac{\partial \theta}{\partial t} &= u_\theta. \end{aligned} \tag{53}$$

Navier-Stokes equations for incompressible flow in Cartesian coordinates, without forcing function, are:

$$\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t}+u_x\frac{\partial u_x}{\partial x}+u_y\frac{\partial u_x}{\partial y}+u_z\frac{\partial u_x}{\partial z}\right)&=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u_x}{\partial x^2}+\frac{\partial^2 u_x}{\partial y^2}+\frac{\partial^2 u_x}{\partial z^2}\right),\\
\rho\left(\frac{\partial u_y}{\partial t}+u_x\frac{\partial u_y}{\partial x}+u_y\frac{\partial u_y}{\partial y}+u_z\frac{\partial u_y}{\partial z}\right)&=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 u_y}{\partial x^2}+\frac{\partial^2 u_y}{\partial y^2}+\frac{\partial^2 u_y}{\partial z^2}\right),\\
\rho\left(\frac{\partial u_z}{\partial t}+u_x\frac{\partial u_z}{\partial x}+u_y\frac{\partial u_z}{\partial y}+u_z\frac{\partial u_z}{\partial z}\right)&=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 u_z}{\partial x^2}+\frac{\partial^2 u_z}{\partial y^2}+\frac{\partial^2 u_z}{\partial z^2}\right).
\end{aligned}\tag{54}$$

And the continuity equation for incompressible flow in Cartesian coordinates is:

$$\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}=0.\tag{55}$$

Using:

$$\begin{aligned}
\frac{\partial r}{\partial z}&=\frac{1}{2}(z^2+y^2)^{-\frac{1}{2}}2z=\frac{z}{\sqrt{z^2+y^2}}=\frac{r\cos(\theta)}{r}=\cos(\theta),\\
\frac{\partial r}{\partial y}&=\frac{1}{2}(z^2+y^2)^{-\frac{1}{2}}2y=\frac{y}{\sqrt{z^2+y^2}}=\frac{r\sin(\theta)}{r}=\sin(\theta),\\
\frac{\partial \theta}{\partial z}&=\frac{1}{1+\frac{y^2}{z^2}}\left(\frac{-y}{z^2}\right)=\frac{-y}{z^2+y^2}=\frac{-r\sin(\theta)}{r^2}=-\frac{1}{r}\sin(\theta),\\
\frac{\partial \theta}{\partial y}&=\frac{1}{1+\frac{y^2}{z^2}}\left(\frac{1}{z}\right)=\frac{z}{z^2+y^2}=\frac{r\cos(\theta)}{r^2}=\frac{1}{r}\cos(\theta).
\end{aligned}\tag{56}$$

And filling equations (52,56) into equation (55) eventually yields:

$$\frac{\partial u_x}{\partial x}+\frac{1}{r}\frac{\partial (ru_r)}{\partial r}+\frac{1}{r}\frac{\partial u_\theta}{\partial \theta}=0.\tag{57}$$

The same routine can be followed for the Navier-Stokes equations, which will not be further specified, see [27] for a reference. Eventually this yields:

$$\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t}+u_r\frac{\partial u_x}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_x}{\partial \theta}+u_x\frac{\partial u_x}{\partial x}\right)+\frac{\partial p}{\partial x}&\\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_x}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2}+\frac{\partial^2 u_x}{\partial x^2}\right)&=0,\\
\rho\left(\frac{\partial u_r}{\partial t}+u_r\frac{\partial u_r}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta}+u_x\frac{\partial u_r}{\partial x}-\frac{u_\theta^2}{r}\right)+\frac{\partial p}{\partial r}&\\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2}+\frac{\partial^2 u_r}{\partial x^2}-\frac{u_r}{r^2}-\frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right)&=0,\\
\rho\left(\frac{\partial u_\theta}{\partial t}+u_r\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta}+u_x\frac{\partial u_\theta}{\partial x}+\frac{u_r u_\theta}{r}\right)+\frac{1}{r}\frac{\partial p}{\partial \theta}&\\
-\mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\theta}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2}+\frac{\partial^2 u_\theta}{\partial x^2}-\frac{u_\theta}{r^2}+\frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right)&=0.
\end{aligned}\tag{58}$$

C Nondimensionalization of the Navier-Stokes equations

For nondimensionalization the variables r^*, u^* are used for nondimensionalization, p will be nondimensionalized by applying $\hat{p} = \frac{p}{\rho(u^*)^2}$, and $\hat{t} = \frac{r^*}{u^*}$. The hat will indicate the nondimensionalization variables. This enables us to rewrite the continuity equation to:

$$\frac{u^*}{r^*}\left(\frac{\partial \hat{u}_x}{\partial \hat{x}} + \frac{1}{\hat{r}}\frac{\partial(\hat{r}\hat{u}_r)}{\partial \hat{r}} + \frac{1}{\hat{r}}\frac{\partial \hat{u}_\theta}{\partial \theta}\right) = 0. \tag{59}$$

And by omitting the subscript and the constant, this can be simply rewritten to the form of equation (4). The same procedure can be followed for the Navier-Stokes equations:

$$\begin{aligned}
&\rho\left(\frac{(u^*)^2}{r^*}\frac{\partial \hat{u}_x}{\partial \hat{t}} + \frac{(u^*)^2}{r^*}\left(\hat{u}_r\frac{\partial \hat{u}_x}{\partial \hat{r}} + \frac{\hat{u}_\theta}{\hat{r}}\frac{\partial \hat{u}_x}{\partial \theta} + \hat{u}_x\frac{\partial \hat{u}_x}{\partial \hat{x}}\right)\right)\\
&+\frac{\rho(u^*)^2}{r^*}\frac{\partial \hat{p}}{\partial \hat{x}} - \mu\frac{u^*}{(r^*)^2}\left(\frac{1}{\hat{r}}\frac{\partial}{\partial \hat{r}}\left(\hat{r}\frac{\partial \hat{u}_x}{\partial \hat{r}}\right) + \frac{1}{\hat{r}^2}\frac{\partial^2 \hat{u}_x}{\partial \theta^2} + \frac{\partial^2 \hat{u}_x}{\partial \hat{x}^2}\right) = 0,\\
&\rho\left(\frac{(u^*)^2}{r^*}\frac{\partial \hat{u}_r}{\partial \hat{t}} + \frac{(u^*)^2}{r^*}\left(\hat{u}_r\frac{\partial \hat{u}_r}{\partial \hat{r}} + \frac{\hat{u}_\theta}{\hat{r}}\frac{\partial \hat{u}_r}{\partial \theta} + \hat{u}_x\frac{\partial \hat{u}_r}{\partial \hat{x}} - \frac{\hat{u}_\theta^2}{\hat{r}}\right)\right)\\
&+\frac{\rho(u^*)^2}{r^*}\frac{\partial \hat{p}}{\partial \hat{r}} - \mu\frac{u^*}{(r^*)^2}\left(\frac{1}{\hat{r}}\frac{\partial}{\partial \hat{r}}\left(\hat{r}\frac{\partial \hat{u}_r}{\partial \hat{r}}\right) + \frac{1}{\hat{r}^2}\frac{\partial^2 \hat{u}_r}{\partial \theta^2} + \frac{\partial^2 \hat{u}_r}{\partial \hat{x}^2} - \frac{\hat{u}_r}{\hat{r}^2} - \frac{2}{\hat{r}^2}\frac{\partial \hat{u}_\theta}{\partial \theta}\right) = 0,\\
&\rho\frac{(u^*)^2}{r^*}\left(\frac{\partial \hat{u}_\theta}{\partial \hat{t}} + \hat{u}_r\frac{\partial \hat{u}_\theta}{\partial \hat{r}} + \frac{\hat{u}_\theta}{\hat{r}}\frac{\partial \hat{u}_\theta}{\partial \theta} + \hat{u}_x\frac{\partial \hat{u}_\theta}{\partial \hat{x}} + \frac{\hat{u}_r\hat{u}_\theta}{\hat{r}}\right)\\
&+\frac{\rho(u^*)^2}{r^*}\frac{1}{\hat{r}}\frac{\partial \hat{p}}{\partial \theta} - \mu\frac{u^*}{(r^*)^2}\left(\frac{1}{\hat{r}}\frac{\partial}{\partial \hat{r}}\left(\hat{r}\frac{\partial \hat{u}_\theta}{\partial \hat{r}}\right) + \frac{1}{\hat{r}^2}\frac{\partial^2 \hat{u}_\theta}{\partial \theta^2} + \frac{\partial^2 \hat{u}_\theta}{\partial \hat{x}^2} - \frac{\hat{u}_\theta}{\hat{r}^2} + \frac{2}{\hat{r}^2}\frac{\partial \hat{u}_r}{\partial \theta}\right) = 0.
\end{aligned} \tag{60}$$

Omitting the hat, this can be simplified to:

$$\begin{aligned}
&\frac{\partial u_x}{\partial t} + u_r\frac{\partial u_x}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_x}{\partial \theta} + u_x\frac{\partial u_x}{\partial x} + \frac{\partial p}{\partial x}\\
&-\frac{\mu}{\rho u^* r^*}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_x}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_x}{\partial \theta^2} + \frac{\partial^2 u_x}{\partial x^2}\right) = 0,\\
&\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} + u_x\frac{\partial u_r}{\partial x} - \frac{u_\theta^2}{r} + \frac{1}{\rho^2}\frac{\partial p}{\partial r}\\
&-\frac{\mu}{\rho u^* r^*}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} + \frac{\partial^2 u_r}{\partial x^2} - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right) = 0,\\
&\frac{\partial u_\theta}{\partial t} + u_r\frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta} + u_x\frac{\partial u_\theta}{\partial x} + \frac{u_r u_\theta}{r} + \frac{1}{\rho^2}\frac{1}{r}\frac{\partial p}{\partial \theta}\\
&-\frac{\mu}{\rho u^* r^*}\left(\frac{1}{r}\frac{\partial}{\partial \hat{r}}\left(r\frac{\partial u_\theta}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} + \frac{\partial^2 u_\theta}{\partial x^2} - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right) = 0.
\end{aligned} \tag{61}$$

By setting $r^* = R_1$ and $u^* = u_{\max}$, this simplifies to:

$$\begin{aligned}
\frac{\partial u_x}{\partial t} + u_r\frac{\partial u_x}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_x}{\partial \theta} + u_x\frac{\partial u_x}{\partial x} + \frac{\partial p}{\partial x} \\
-\frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_x}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_x}{\partial \theta^2} + \frac{\partial^2 u_x}{\partial x^2}\right) = 0, \\
\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} + u_x\frac{\partial u_r}{\partial x} - \frac{u_\theta^2}{r} + \frac{\partial p}{\partial r} \\
-\frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} + \frac{\partial^2 u_r}{\partial x^2} - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right) = 0, \\
\frac{\partial u_\theta}{\partial t} + u_r\frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta} + u_x\frac{\partial u_\theta}{\partial x} + \frac{u_r u_\theta}{r} + \frac{1}{r}\frac{\partial p}{\partial \theta} \\
-\frac{1}{Re}\left(\frac{1}{r}\frac{\partial}{\partial \hat{r}}\left(r\frac{\partial u_\theta}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} + \frac{\partial^2 u_\theta}{\partial x^2} - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right) = 0.
\end{aligned}$$

In vector notation, using the continuity equation $-\nabla\cdot\mathbf{u} = 0$, this can be rewritten towards:

$$\begin{aligned}
\frac{\partial u_x}{\partial t} + \nabla\cdot(u_x\mathbf{u}) + \frac{\partial p}{\partial x} - \frac{1}{Re}\nabla^2 u_x = 0, \qquad (62)\\
\frac{\partial u_r}{\partial t} + \nabla\cdot(u_r\mathbf{u}) - \frac{u_\theta^2}{r} + \frac{\partial p}{\partial r} - \frac{1}{Re}\left(\nabla^2 u_r - \frac{u_r}{r^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right) = 0, \\
\frac{\partial u_\theta}{\partial t} + \nabla\cdot(u_\theta\mathbf{u}) + \frac{u_r u_\theta}{r} + \frac{1}{r}\frac{\partial p}{\partial \theta} - \frac{1}{Re}\left(\nabla^2 u_\theta - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right) = 0.
\end{aligned}$$

Adding the stability term $-\frac{1}{2}(\nabla\cdot\mathbf{u})\mathbf{u}$ from [15] and omitting gravity eventually leads to equation (7).

D Elements of the spectral collocation matrix

The diagonal elements of the collocation differentiation matrix are given by [28]:

$$\begin{aligned}
(D_N)_{j,m} &= \frac{\pi}{NL}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} \imath k \exp\left(\frac{2\pi\imath k}{N}(j-m)\right), \\
&= \frac{\pi}{NL}\sum_{k=1}^{\frac{N}{2}-1} \imath k\left(\exp\left(\frac{2\pi\imath k}{N}(j-m)\right) - \exp\left(-\frac{2\pi\imath k}{N}(j-m)\right)\right), \qquad (63)\\
&= \frac{\pi}{NL}\sum_{k=1}^{\frac{N}{2}-1} -2k\sin\left(\frac{2\pi k}{N}(j-m)\right).
\end{aligned}$$

The diagonal entries are 0: $(D_N)_{j,j}$, as can be verified easily by noting $\sin(0) = 0$. For the off-diagonal entries, the derivation is more complex, this will now follow. Using the equality [26]:

$$\sum_{k=1}^{n} k\sin(ka) = \frac{1}{4}\csc^2\left(\frac{a}{2}\right)((n+1)\sin(na) - n\sin((n+1)a). \qquad (64)$$

Equation (63) simplifies to:

$$\begin{aligned}
&= -\frac{2\pi}{NL}\frac{1}{4}\csc^2\left(\frac{\pi}{N}(j-m)\right)\left(\frac{N}{2}\sin\left(\frac{2\pi}{N}\left(\frac{N}{2}-1\right)(j-m)\right)\right.\\
&\quad \left.-\left(\frac{N}{2}-1\right)\sin\left(\frac{N}{2}\frac{2\pi}{N}(j-m)\right)\right)\\
&= -\frac{\pi}{2NL}\csc^2\left(\frac{\pi}{N}(j-m)\right)\left(\frac{N}{2}\sin\left(\left(\pi-\frac{2\pi}{N}\right)(j-m)\right)\right.\\
&\quad \left.-\left(\frac{N}{2}-1\right)\sin(\pi(j-m))\right).
\end{aligned} \tag{65}$$

Here we note for $(j-m)\in\mathbb{N}$:

$$\sin(\pi(j-m)) = 0. \tag{66}$$

We use:

$$\sin(2a) = 2\sin(a)\cos(a). \tag{67}$$

And:

$$\sin(a-b) = \sin(a)\cos(b) - \cos(a)\sin(b). \tag{68}$$

Then Equation (65) simplifies to:

$$\begin{aligned}
&= -\frac{\pi}{2NL}\csc^2\left(\frac{\pi}{N}(j-m)\right)\frac{N}{2}\sin\left(\left(\pi-\frac{2\pi}{N}\right)(j-m)\right)\\
&= -\frac{\pi}{4L}\csc^2\left(\frac{\pi}{N}(j-m)\right)\left(\sin(\pi(j-m))\cos\left(\frac{2\pi}{N}(j-m)\right)\right.\\
&\quad \left.-\cos(\pi(j-m))\sin\left(\frac{2\pi}{N}(j-m)\right)\right)\\
&= \frac{\pi}{4L}\frac{1}{\sin^2\left(\frac{\pi}{N}(j-m)\right)}\cos(\pi(j-m))\sin\left(\frac{2\pi}{N}(j-m)\right)\\
&= \frac{\pi}{4L}\frac{1}{\sin^2\left(\frac{\pi}{N}(j-m)\right)}\cos(\pi(j-m))2\sin\left(\frac{\pi}{N}(j-m)\right)\cos\left(\frac{\pi}{N}(j-m)\right)\\
&= \frac{\pi}{2L}\frac{\cos\left(\frac{\pi}{N}(j-m)\right)}{\sin\left(\frac{\pi}{N}(j-m)\right)}\cos(\pi(j-m))\\
&= \frac{\pi}{2L}\cot\left(\frac{\pi}{N}(j-m)\right)\cos(\pi(j-m)).
\end{aligned} \tag{69}$$

For $k\in\mathbb{N}$:

$$\begin{aligned}
\cos\left((1+2k)\pi\right) &= -1,\\
\cos(2k\pi) &= 1.
\end{aligned} \tag{70}$$

Therefore for $j,m\in\mathbb{N}_{\text{even}}$ and for $j,m\in\mathbb{N}_{\text{odd}}$ $\cos(\pi(j-m)) = 1$. In this case $j+m$ will be even. For $j\in\mathbb{N}_{\text{odd}}, m\in\mathbb{N}_{\text{even}}$ and $j\in\mathbb{N}_{\text{even}}, m\in\mathbb{N}_{\text{odd}}$ $\cos(\pi(j-m)) = -1$. In this case $j+m$ will be odd. Considering $a\in\mathbb{N}$:

$$\begin{aligned}
(-1)^{2a} &= 1,\\
(-1)^{2a+1} &= -1.
\end{aligned} \tag{71}$$

Equation (63) can then finally be rewritten to:

$$(D_N)_{j,m} = (-1)^{j+m}\frac{\pi}{2L}\cot\left(\frac{\pi}{N}(j-m)\right) \tag{72}$$

E Spectral discretisation of the azimuthal direction

Starting from the general incompressible non-dimensionalized Naviers-Stokes equations:

$$\begin{aligned} \mathbf{u}\cdot\nabla\mathbf{u}+\nabla p-\frac{1}{Re}\nabla\cdot\nabla\mathbf{u}=\mathbf{0},\\ \nabla\cdot\mathbf{u}=0. \end{aligned} \tag{73}$$

This can be written in the weak form (assuming $\mathbf{v}|_{\mathrm{d}\Omega}$, $\mathrm{d}\Omega = r\mathrm{d}x\mathrm{d}r\mathrm{d}\theta$)as:

$$\begin{aligned} \int_\Omega (\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{v}+\nabla p\cdot\mathbf{v}-\frac{1}{Re}(\nabla\cdot\nabla\mathbf{u})\cdot\mathbf{v}\mathrm{d}\Omega=0\\ =\int_\Omega (\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{v}-p\nabla\cdot\mathbf{v}+\frac{1}{Re}\nabla\mathbf{u}:\nabla\mathbf{v}\mathrm{d}\Omega+\int_{\partial\Omega}p(\nu\cdot\mathbf{v})-\frac{1}{Re}(\nabla\mathbf{u}\cdot\nu)\cdot\mathbf{v}\mathrm{d}S. \end{aligned} \tag{74}$$

Resulting in:

$$\begin{aligned} \int_\Omega (\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{v}-p\nabla\cdot\mathbf{v}+\frac{1}{Re}\nabla\mathbf{u}:\nabla\mathbf{v}\mathrm{d}\Omega=0,\\ \int_\Omega \nabla\cdot\mathbf{u}q\mathrm{d}\Omega=0. \end{aligned} \tag{75}$$

The gradient of $\mathbf{u}$ in cylindrical coordinates is given by:

$$\nabla\mathbf{u}=\begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial r} & \frac{1}{r}\frac{\partial u_x}{\partial\theta}\\ \frac{\partial u_r}{\partial x} & \frac{\partial u_r}{\partial r} & \frac{1}{r}(\frac{\partial u_r}{\partial\theta}-u_\theta)\\ \frac{\partial u_\theta}{\partial x} & \frac{\partial u_\theta}{\partial r} & \frac{1}{r}(\frac{\partial u_\theta}{\partial\theta}+u_r) \end{pmatrix}. \tag{76}$$

The gradient of p (scalar) in cylindrical coordinates is given by:

$$\nabla p=\frac{\partial p}{\partial x}+\frac{\partial p}{\partial r}+\frac{1}{r}\frac{\partial p}{\partial\theta}. \tag{77}$$

The divergence of $\mathbf{v}$ in cylindrical coordinates is given by:

$$\nabla\cdot\mathbf{v}=\frac{\partial v_x}{\partial x}+\frac{\partial v_r}{\partial r}+\frac{1}{r}\frac{\partial v_\theta}{\partial\theta}+\frac{v_r}{r}. \tag{78}$$

The trapezium rule approximates an integral on an interval $[a,b]$ as:

$$\int_a^b f(x)dx\approx\frac{(b-a)}{2}(f(b)+f(a)). \tag{79}$$

For an equally spaced grid, with N grid points, $x_1=a$, $x_{N+1}=b$ and constant grid spacing this results in:

$$\int_a^b f(x)dx\approx\frac{b-a}{2N}(f(b)+f(a)+2\sum_{i=2}^{N}f(x_i)). \tag{80}$$

For a periodic function, with $f(x_1)=f(x_{N+1})$, this simplifies to:

$$\int_a^b f(x)dx\approx\frac{b-a}{N}\sum_{i=1}^{N}f(x_i). \tag{81}$$

For the azimuthal direction, with period 2π, this can be written as:

$$\int_0^{2\pi} f(x)dx \approx \frac{2\pi}{N}\sum_{i=1}^{N} f(x_i). \tag{82}$$

The inner product of two matrices $A = (a_{ij})$ and $B = (b_{ij})$ can be calculated using:

$$A : B = \mathrm{tr}(A^\intercal B) = a_{11}b_{11}+a_{12}b_{12}+a_{13}b_{13}+a_{21}b_{21}+a_{22}b_{22}+a_{23}b_{23}+a_{31}b_{31}+a_{32}b_{32}+a_{33}b_{33}. \tag{83}$$

The Navier-Stokes equations can be rewritten as:

$$\begin{aligned}
&\int_\Omega \begin{pmatrix} u_x\frac{\partial u_x}{\partial x} + u_r\frac{\partial u_x}{\partial r} + u_\theta\frac{1}{r}\frac{\partial u_x}{\partial \theta} \\ u_x\frac{\partial u_r}{\partial x} + u_r\frac{\partial u_r}{\partial r} + u_\theta\frac{1}{r}(\frac{\partial u_r}{\partial \theta} - u_\theta) \\ u_x\frac{\partial u_\theta}{\partial x} + u_r\frac{\partial u_\theta}{\partial r} + u_\theta\frac{1}{r}(\frac{\partial u_\theta}{\partial \theta} + u_r) \end{pmatrix}\cdot\mathbf{v} - p\nabla\cdot\mathbf{v} \\
&+\frac{1}{Re}\begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial r} & \frac{1}{r}\frac{\partial u_x}{\partial \theta} \\ \frac{\partial u_r}{\partial x} & \frac{\partial u_r}{\partial r} & \frac{1}{r}(\frac{\partial u_r}{\partial \theta} - u_\theta) \\ \frac{\partial u_\theta}{\partial x} & \frac{\partial u_\theta}{\partial r} & \frac{1}{r}(\frac{\partial u_\theta}{\partial \theta} + u_r) \end{pmatrix} : \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial r} & \frac{1}{r}\frac{\partial v_x}{\partial \theta} \\ \frac{\partial v_r}{\partial x} & \frac{\partial v_r}{\partial r} & \frac{1}{r}(\frac{\partial v_r}{\partial \theta} - v_\theta) \\ \frac{\partial u_\theta}{\partial x} & \frac{\partial v_\theta}{\partial r} & \frac{1}{r}(\frac{\partial v_\theta}{\partial \theta} + v_r) \end{pmatrix}\mathrm{d}\Omega = 0, \\
&\int_\Omega \nabla\cdot\mathbf{u}q\mathrm{d}\Omega = 0.
\end{aligned} \tag{84}$$

$$\begin{aligned}
&\int_\Omega \left(u_x\frac{\partial u_x}{\partial x} + u_r\frac{\partial u_x}{\partial r} + u_\theta\frac{1}{r}\frac{\partial u_x}{\partial \theta}\right)v_x + \left(u_x\frac{\partial u_r}{\partial x} + u_r\frac{\partial u_r}{\partial r} + u_\theta\frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta\right)\right)v_r \\
&+\left(u_x\frac{\partial u_\theta}{\partial x} + u_r\frac{\partial u_\theta}{\partial r} + u_\theta\frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r\right)\right)v_\theta - p\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}\right) \\
&+\frac{1}{Re}\left(\frac{\partial u_x}{\partial x}\frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial r}\frac{\partial v_x}{\partial r} + \frac{1}{r}\frac{\partial u_x}{\partial \theta}\frac{1}{r}\frac{\partial v_x}{\partial \theta}\right. \\
&+\frac{\partial u_r}{\partial x}\frac{\partial v_r}{\partial x} + \frac{\partial u_r}{\partial r}\frac{\partial v_r}{\partial r} + \frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta\right)\frac{1}{r}\left(\frac{\partial v_r}{\partial \theta} - v_\theta\right) \\
&\left.+\frac{\partial u_\theta}{\partial x}\frac{\partial v_\theta}{\partial x} + \frac{\partial u_\theta}{\partial r}\frac{\partial v_\theta}{\partial r} + \frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r\right)\frac{1}{r}\left(\frac{\partial v_\theta}{\partial \theta} + v_r\right)\right)\mathrm{d}\Omega = 0, \\
&\int_\Omega \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_r}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r}\right)q\mathrm{d}\Omega = 0.
\end{aligned} \tag{85}$$

Discretisation in the θ-direction then yields:

$$\int_{x,r} \frac{2\pi}{N} \sum_{i=1}^{N} \left(\left(U_x^i \frac{\partial U_x^i}{\partial x} + U_r^i \frac{\partial U_x^i}{\partial r} + U_\theta^i \frac{1}{r} \sum_{j=1}^{N} D_\theta^{ij} U_x^j \right) V_x^i \right.$$
$$+ \left(U_x^i \frac{\partial U_r^i}{\partial x} + U_r^i \frac{\partial U_r^i}{\partial r} + U_\theta^i \frac{1}{r} \left(\sum_{j=1}^{N} D_\theta^{ij} U_r^j - U_\theta^i \right) \right) V_r^i$$
$$+ \left(U_x^i \frac{\partial U_\theta^i}{\partial x} + U_r^i \frac{\partial U_\theta^i}{\partial r} + U_\theta^i \frac{1}{r} \left(\sum_{j=1}^{N} D_\theta^{ij} U_\theta^j + U_r^i \right) \right) V_\theta^i$$
$$- P^i \left(\frac{\partial V_x^i}{\partial x} + \frac{\partial V_r^i}{\partial r} + \frac{1}{r} \sum_{j=1}^{N} D_\theta^{ij} V_\theta^j + \frac{V_r^i}{r} \right)$$
$$+ \frac{1}{Re} \left(\frac{\partial U_x^i}{\partial x} \frac{\partial V_x^i}{\partial x} + \frac{\partial U_x^i}{\partial r} \frac{\partial V_x^i}{\partial r} + \frac{1}{r^2} \sum_{j=1}^{N} D_\theta^{ij} U_x^j \sum_{j=1}^{N} D_\theta^{ij} V_x^j \right.$$
$$+ \frac{\partial U_r^i}{\partial x} \frac{\partial V_r^i}{\partial x} + \frac{\partial U_r^i}{\partial r} \frac{\partial V_r{}^i}{\partial r} + \frac{1}{r^2} \left(\sum_{j=1}^{N} D_\theta^{ij} U_r^j - U_\theta^i \right) \left(\sum_{j=1}^{N} D_\theta^{ij} V_r^j - V_\theta^i \right)$$
$$\left. \left. + \frac{\partial U_\theta^i}{\partial x} \frac{\partial V_\theta^i}{\partial x} + \frac{\partial U_\theta^i}{\partial r} \frac{\partial V_\theta^i}{\partial r} + \frac{1}{r^2} \left(\sum_{j=1}^{N} D_\theta^{ij} U_\theta^j + U_r^i \right) \left(\sum_{j=1}^{N} D_\theta^{ij} V_\theta^j + V_r^i \right) \right) \right) r \mathrm{d}r \mathrm{d}x = 0,$$
$$\int_{x,r} \frac{2\pi}{N} \sum_{i=1}^{N} \left(\frac{\partial U_x^i}{\partial x} + \frac{\partial U_r^i}{\partial r} + \frac{1}{r} \sum_{j=1}^{N} D_\theta^{ij} U_\theta^j + \frac{U_r^i}{r} \right) q^i r \mathrm{d}r \mathrm{d}x = 0. \tag{86}$$

Here D_θ is the Fourier spectral collocation matrix with respect to θ, U_x, U_r and U_θ are velocity column vectors of length N, P is the pressure column vector, V_x, V_r and V_θ are vectors of the test functions.

$D_\theta^\intercal = -D_\theta$: it is skew-symmetric.

$$\begin{aligned} &\sum_{i=1}^{N} \sum_{j=1}^{N} D_\theta^{ij} U^j \sum_{k=1}^{N} D_\theta^{ik} U^k = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} D_\theta^{ij} U^j D_\theta^{ik} U^k \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} (D_\theta^{ik} U^k)^\intercal D_\theta^{ij} U^j = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} -U^{k\intercal} D_\theta^{ki} D_\theta^{ij} U^j \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} -U^{k\intercal} D_\theta^{2i,kj} U^j \end{aligned} \tag{87}$$

Equation (86) then yields:

$$\int_{x,r}\left(\frac{2\pi}{N}\sum_{i=1}^{N}\left(\left(\left(U_x^i\frac{\partial U_x^i}{\partial x}+U_r^i\frac{\partial U_x^i}{\partial r}+U_\theta^i\frac{1}{r}\sum_{j=1}^{N}D_\theta^{ij}U_x^j\right)V_x^i\right.\right.\right.$$
$$+\left(U_x^i\frac{\partial U_r^i}{\partial x}+U_r^i\frac{\partial U_r^i}{\partial r}+U_\theta^i\frac{1}{r}\left(\sum_{j=1}^{N}D_\theta^{ij}U_r^j-U_\theta^i\right)\right)V_r^i$$
$$+\left(U_x^i\frac{\partial U_\theta^i}{\partial x}+U_r^i\frac{\partial U_\theta^i}{\partial r}+U_\theta^i\frac{1}{r}\left(\sum_{j=1}^{N}D_\theta^{ij}U_\theta^j+U_r^i\right)\right)V_\theta^i$$
$$-P^i\left(\frac{\partial V_x^i}{\partial x}+\frac{\partial V_r^i}{\partial r}+\frac{1}{r}\sum_{j=1}^{N}D_\theta^{ij}V_\theta^j+\frac{V_r^i}{r}\right)$$
$$+\frac{1}{Re}\left(\frac{\partial U_x^i}{\partial x}\frac{\partial V_x^i}{\partial x}+\frac{\partial U_x^i}{\partial r}\frac{\partial V_x^i}{\partial r}-\frac{1}{r^2}\sum_{j=1}^{N}\sum_{k=1}^{N}D_\theta^{2,kj}U_x^jV_x^k+\frac{\partial U_r^i}{\partial x}\frac{\partial V_r^i}{\partial x}\right.$$
$$\left.+\frac{\partial U_r^i}{\partial r}\frac{\partial V_r{}^i}{\partial r}+\frac{\partial U_\theta^i}{\partial x}\frac{\partial V_\theta^i}{\partial x}+\frac{\partial U_\theta^i}{\partial r}\frac{\partial V_\theta^i}{\partial r}\right.$$
$$+\frac{1}{r^2}\left(-\sum_{j=1}^{N}D_\theta^{ij}U_r^jV_\theta^i+U_\theta^iV_\theta^i+\sum_{j=1}^{N}V_r^jD_\theta^{ij}U_\theta^i-\sum_{j=1}^{N}\sum_{k=1}^{N}V_r^kD_\theta^{2,kj}U_r^j\right)$$
$$\left.\left.\left.+\frac{1}{r^2}\left(\sum_{j=1}^{N}D_\theta^{ij}U_\theta^jV_r^i+U_r^iV_r^i-\sum_{j=1}^{N}V_\theta^jD_\theta^{ij}U_r^i-\sum_{j=1}^{N}\sum_{k=1}^{N}D_\theta^{2,kj}U_\theta^jV_\theta^k\right)\right)\right)rdrdx=0,$$
$$\int_{x,r}\frac{2\pi}{N}\sum_{i=1}^{N}\left(\frac{\partial U_x^i}{\partial x}+\frac{\partial U_r^i}{\partial r}+\frac{1}{r}\sum_{j=1}^{N}D_\theta^{ij}U_\theta^j+\frac{U_r^i}{r}\right)q^irdrdx=0. \tag{88}$$

F Python code for generating the Fortran90 spectral collocation input

This code was written by Andrew Cliffe:

```
# Put a general descritpion here:
#
# Navier Stokes equations in 2d cartesian coordinates. Strain-rate dependent
    viscosity.
#
# Select code
Code = 'AptoFEM' # Choose from ['ENTWIFE', 'AptoFEM']

# Output file name
OutputFile = 'Pipe10.f90'

# The following are integer parameters
```

```
numSpaceDimensions = 2 # Number of space dimensions
numVariables = 40       # Number of variables

#The following is a list of parameter name / parameter number pairs.
ParameterNames = [('REYNOLDS NUMBER',1),]
#ParameterNames = [('REYNOLDS NUMBER',1),('ALPHA',2),('N',3)]

# The next parameter is a string
derivatives = 'Only continuation' # Choose from ['Only Jacobian', 'Up to 1st
    derviatives', 'Up to 2nd derivatives', 'Only continuation', 'All']
#derivatives = 'Up to 1st derivatives'

# The next parameters are booleans. Note: they cannot both be set to False.
steadyState = True
transient = False

# The following is a list pointing to files in the current directory and
    giving their region number and part number
UserEquations = [('Pipe.py',1,1)]

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------

USER EQUATION

-------------------------------------------------------------------------------
File = Pipe.py
Type = User Equation
Region = 1
Part = 1
-------------------------------------------------------------------------------
def Trace(A):
      return A.trace()
def dot(vecA,vecB):
      return (vecA.T*vecB).trace()
from sympy import var as symvar
from sympy import zeros
from sympy import N

symvar('nc')
nc = 10

Re = pvals[1]

R = r[2]

# Calculate spectral differentiation matrix
# D = Matrix(nc,nc,lambda i,j: 0 if i==j else (cot( (i-j)*pi/nc
    )*(-1)**abs(i-j)/2) )
D = zeros(nc)
D2 = zeros(nc)
for i in range(nc):
      for j in range(nc):
```

```
            D[i,j] = N(cot( (i-j)*pi/nc )*(-1)**abs(i-j)/2)
      D[i,i] = 0
D2 = D*D

# Set up variables and their derivatives
ux = zeros((nc,1))
ur = zeros((nc,1))
ut = zeros((nc,1))
p = zeros((nc,1))
duxdx = zeros((nc,1))
duxdr = zeros((nc,1))
duxdt = zeros((nc,1))
durdx = zeros((nc,1))
durdr = zeros((nc,1))
durdt = zeros((nc,1))
dutdx = zeros((nc,1))
dutdr = zeros((nc,1))
dutdt = zeros((nc,1))
for i in range(nc):
      ux[i] = var[i+1]
      ur[i] = var[i+nc+1]
      ut[i] = var[i+2*nc+1]
      p[i] = var[i+3*nc+1]
      duxdx[i] = dvdr[i+1,1]
      duxdr[i] = dvdr[i+1,2]
      durdx[i] = dvdr[i+nc+1,1]
      durdr[i] = dvdr[i+nc+1,2]
      dutdx[i] = dvdr[i+2*nc+1,1]
      dutdr[i] = dvdr[i+2*nc+1,2]
duxdt = D*ux
durdt = D*ur
dutdt = D*ut
d2uxdt2 = D2*ux
d2urdt2 = D2*ur
d2utdt2 = D2*ut

# Set up test functions and their derivatives
Nux = zeros((nc,1))
Nur = zeros((nc,1))
Nut = zeros((nc,1))
Np = zeros((nc,1))
dNuxdx = zeros((nc,1))
dNuxdr = zeros((nc,1))
dNuxdt = zeros((nc,1))
dNurdx = zeros((nc,1))
dNurdr = zeros((nc,1))
dNurdt = zeros((nc,1))
dNutdx = zeros((nc,1))
dNutdr = zeros((nc,1))
dNutdt = zeros((nc,1))
for i in range(nc):
      Nux[i] = n[i+1]
      Nur[i] = n[i+nc+1]
```

```
        Nut[i] = n[i+2*nc+1]
        Np[i] = n[i+3*nc+1]
        dNuxdx[i] = dndr[i+1,1]
        dNuxdr[i] = dndr[i+1,2]
        dNurdx[i] = dndr[i+nc+1,1]
        dNurdr[i] = dndr[i+nc+1,2]
        dNutdx[i] = dndr[i+2*nc+1,1]
        dNutdr[i] = dndr[i+2*nc+1,2]
dNuxdt = D*Nux
dNurdt = D*Nur
dNutdt = D*Nut

# Cacluate weak form
weakF = 0
for i in range(nc):
        U = Matrix( [ [ux[i]],[ur[i]],[ut[i]] ] )
        gradU = Matrix( [ [ duxdx[i], duxdr[i], duxdt[i]/R ],
                                        [ durdx[i], durdr[i], (durdt[i]-ut[i])/R
                                            ],
                                        [ dutdx[i], dutdr[i], (dutdt[i]+ur[i])/R
                                            ] ] )
        divU = Trace(gradU)
        NU = Matrix( [ [Nux[i]],[Nur[i]],[Nut[i]] ] )
        gradNU = Matrix( [ [ dNuxdx[i], dNuxdr[i], dNuxdt[i]/R ],
                                         [ dNurdx[i], dNurdr[i],
                                             (dNurdt[i]-Nut[i])/R ],
                                         [ dNutdx[i], dNutdr[i],
                                             (dNutdt[i]+Nur[i])/R ] ] )
        visc = duxdx[i]*dNuxdx[i] + durdx[i]*dNurdx[i] + dutdx[i]*dNutdx[i] \
              + duxdr[i]*dNuxdr[i] + durdr[i]*dNurdr[i] + dutdr[i]*dNutdr[i] \
              - d2uxdt2[i]*Nux[i]/R**2 \
              - (d2urdt2[i]-dutdt[i])*Nur[i]/R**2 - (durdt[i]-ut[i])*Nut[i]/R**2
                  \
              - (d2utdt2[i]+durdt[i])*Nut[i]/R**2+(dutdt[i]+ur[i])*Nur[i]/R**2
        divNU = Trace(gradNU)
        mom = R*( dot( (gradU*U), NU ) - p[i]*divNU + visc/Re )
        cont = R*( Np[i]*divU )
        weakF = weakF + mom + cont
```

G O(2) symmetry

O(2) is the Lie group generated by rotations and reflections. Every compact Lie group in a more general sense is topologically isomorphic to the linear Lie group [23]. A linear Lie group is a closed subgroup of the group of all invertible linear transformations [23]. The pipe geometry is O(2) symmetric.

According to Golubitsky *et al.* [23, p. 3], for $f : \mathbb{R}^n \to \mathbb{R}^n$ smooth, an ordinary differential equation (ODE):

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) \tag{89}$$

has a symmetry γ, an invertible $n \times n$ matrix, if ($\forall x \in \mathbb{R}^n$):

$$f(\gamma x) = \gamma f(x). \tag{90}$$

Since in this research, the solutions are assumed to be steady-state, the $\frac{\mathrm{d}x}{\mathrm{d}t} = 0$.

Given V a vector space, Γ a compact Lie group, $g : V \to V$ is Γ-equivariant if

$$g(\gamma x) = \gamma g(x) \tag{91}$$

$\forall \gamma \in \Gamma, x \in V$ [23, p. 49].

The O(2) symmetry consists of rotations and reflections through the x-axis. The pipe with sudden expansion is clearly O(2) symmetric in the continuous case with symmetric inflow conditions. For the discretized pipe, the O(2) symmetry for symmetric inflow conditions has yet to be shown. This will be shown by multiplying by ρ_τ, a rotation of τ radians about the x-axis, and σ, a reflection through the x-axis. If the Equation (85) satisfy these operations, they will also satisfy a combination of both (as $\rho_\tau \sigma = \rho_{(\tau+\pi)\mathrm{mod}(2\pi)}$. The O(2) symmetry will hold for the non-discretized equations, but not after discretisation. (The rotations will be limited to $\frac{(2\pi i)\mathrm{mod}(2\pi)}{N}$, with $i \in \mathbb{Z}$ and N the number of collocation points. This will not satisfy the $\forall \gamma \in \Gamma$ constraint in Equation (91).)

Since the functions, g, are 2π-periodic in θ: $g((\theta + \tau)\mathrm{mod}(2\pi)) = g(\theta + \tau)$.

For $\sigma \in$ O(2):

$$\begin{aligned} \sigma(x, r, \theta, p) =& (x, r, (\theta + \pi)\mathrm{mod}(2\pi), p), \\ \sigma(u_x, u_r, u_\theta) =& (u_x, u_r, u_{(\theta+\pi)\mathrm{mod}(2\pi)}), \\ \sigma(v_x, v_r, v_\theta) =& (v_x, v_r, v_{(\theta+\pi)\mathrm{mod}(2\pi)}). \end{aligned} \tag{92}$$

For $\rho_\tau \in$ O(2):

$$\begin{aligned} \sigma(x, r, \theta, p) =& (x, r, (\theta + \pi)\mathrm{mod}(2\tau), p), \\ \sigma(u_x, u_r, u_\theta) =& (u_x, u_r, u_{(\theta+\tau)\mathrm{mod}(2\pi)}), \\ \sigma(v_x, v_r, v_\theta) =& (v_x, v_r, v_{(\theta+\tau)\mathrm{mod}(2\pi)}). \end{aligned} \tag{93}$$

Here mod indicates the modulo operation. Since this will be shown for general τ, if it holds for ρ_τ, it will also hold for $\sigma = \rho_\pi$.

Since the $\int_0^{2\pi} g(\theta)\mathrm{d}\theta = \int_{0+\tau}^{2\pi+\tau} g(\theta)\mathrm{d}\theta$, for $\tau \in \mathbb{R}$, g a continuous 2π-periodic function, rotation and reflection for Equation (91) will be the same as for:

$$\begin{aligned} &\left(u_x \frac{\partial u_x}{\partial x} + u_r \frac{\partial u_x}{\partial r} + u_\theta \frac{1}{r} \frac{\partial u_x}{\partial \theta}\right) v_x + \left(u_x \frac{\partial u_r}{\partial x} + u_r \frac{\partial u_r}{\partial r} + u_\theta \frac{1}{r} \left(\frac{\partial u_r}{\partial \theta} - u_\theta\right)\right) v_r \\ &+ \left(u_x \frac{\partial u_\theta}{\partial x} + u_r \frac{\partial u_\theta}{\partial r} + u_\theta \frac{1}{r} \left(\frac{\partial u_\theta}{\partial \theta} + u_r\right)\right) v_\theta - p \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}\right) \\ &+ \frac{1}{Re} \left(\frac{\partial u_x}{\partial x} \frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial r} \frac{\partial v_x}{\partial r} + \frac{1}{r} \frac{\partial u_x}{\partial \theta} \frac{1}{r} \frac{\partial v_x}{\partial \theta}\right. \\ &+ \frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial x} + \frac{\partial u_r}{\partial r} \frac{\partial v_r}{\partial r} + \frac{1}{r} \left(\frac{\partial u_r}{\partial \theta} - u_\theta\right) \frac{1}{r} \left(\frac{\partial v_r}{\partial \theta} - v_\theta\right) \\ &\left. + \frac{\partial u_\theta}{\partial x} \frac{\partial v_\theta}{\partial x} + \frac{\partial u_\theta}{\partial r} \frac{\partial v_\theta}{\partial r} + \frac{1}{r} \left(\frac{\partial u_\theta}{\partial \theta} + u_r\right) \frac{1}{r} \left(\frac{\partial v_\theta}{\partial \theta} + v_r\right)\right) \\ &\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_r}{\partial r} + \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r}\right) q = 0. \end{aligned} \tag{94}$$

First the continuity equation will be checked:

$$\begin{aligned}
&\rho_\tau \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_r}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \right) q = 0 \\
&= \left(\rho_\tau \frac{\partial u_x}{\partial x} + \rho_\tau \frac{\partial u_r}{\partial r} + \rho_\tau \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \right) q \\
&= \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_r}{\partial r} + \frac{1}{r}\frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} + \frac{u_r}{r} \right) q.
\end{aligned} \tag{95}$$

Equation (91) holds for the continuity equation.
Now for the momentum equations:

$$\begin{aligned}
&\rho_\tau \left(\left(u_x \frac{\partial u_x}{\partial x} + u_r \frac{\partial u_x}{\partial r} + u_\theta \frac{1}{r}\frac{\partial u_x}{\partial \theta} \right) v_x + \left(u_x \frac{\partial u_r}{\partial x} + u_r \frac{\partial u_r}{\partial r} + u_\theta \frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta \right) \right) v_r \right. \\
&+ \left. \left(u_x \frac{\partial u_\theta}{\partial x} + u_r \frac{\partial u_\theta}{\partial r} + u_\theta \frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r \right) \right) v_\theta - p \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} \right) \right. \\
&+ \frac{1}{Re} \left(\frac{\partial u_x}{\partial x}\frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial r}\frac{\partial v_x}{\partial r} + \frac{1}{r}\frac{\partial u_x}{\partial \theta}\frac{1}{r}\frac{\partial v_x}{\partial \theta} \right. \\
&+ \left. \frac{\partial u_r}{\partial x}\frac{\partial v_r}{\partial x} + \frac{\partial u_r}{\partial r}\frac{\partial v_r}{\partial r} + \frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta \right)\frac{1}{r}\left(\frac{\partial v_r}{\partial \theta} - v_\theta \right) \right. \\
&+ \left. \left. \frac{\partial u_\theta}{\partial x}\frac{\partial v_\theta}{\partial x} + \frac{\partial u_\theta}{\partial r}\frac{\partial v_\theta}{\partial r} + \frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r \right)\frac{1}{r}\left(\frac{\partial v_\theta}{\partial \theta} + v_r \right) \right) \right) \\
&= \rho_\tau \left(u_x \frac{\partial u_x}{\partial x} + u_r \frac{\partial u_x}{\partial r} + u_\theta \frac{1}{r}\frac{\partial u_x}{\partial \theta} \right) v_x + \rho_\tau \left(u_x \frac{\partial u_r}{\partial x} + u_r \frac{\partial u_r}{\partial r} + u_\theta \frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta \right) \right) v_r \\
&+ \rho_\tau \left(u_x \frac{\partial u_\theta}{\partial x} + u_r \frac{\partial u_\theta}{\partial r} + u_\theta \frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r \right) \right) v_\theta - \rho_\tau p \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} \right) \\
&+ \frac{1}{Re}\rho_\tau \left(\frac{\partial u_x}{\partial x}\frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial r}\frac{\partial v_x}{\partial r} + \frac{1}{r}\frac{\partial u_x}{\partial \theta}\frac{1}{r}\frac{\partial v_x}{\partial \theta} \right. \\
&+ \left. \frac{\partial u_r}{\partial x}\frac{\partial v_r}{\partial x} + \frac{\partial u_r}{\partial r}\frac{\partial v_r}{\partial r} + \frac{1}{r}\left(\frac{\partial u_r}{\partial \theta} - u_\theta \right)\frac{1}{r}\left(\frac{\partial v_r}{\partial \theta} - v_\theta \right) \right. \\
&+ \left. \left. \frac{\partial u_\theta}{\partial x}\frac{\partial v_\theta}{\partial x} + \frac{\partial u_\theta}{\partial r}\frac{\partial v_\theta}{\partial r} + \frac{1}{r}\left(\frac{\partial u_\theta}{\partial \theta} + u_r \right)\frac{1}{r}\left(\frac{\partial v_\theta}{\partial \theta} + v_r \right) \right)
\end{aligned} \tag{96}$$

$$
\begin{aligned}
&= \left(u_x \frac{\partial u_x}{\partial x} + u_r \frac{\partial u_x}{\partial r} + u_{(\theta+\tau)\mathrm{mod}(2\pi)} \frac{1}{r} \frac{\partial u_x}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} \right) v_x \\
&+ \left(u_x \frac{\partial u_r}{\partial x} + u_r \frac{\partial u_r}{\partial r} + u_{(\theta+\tau)\mathrm{mod}(2\pi)} \frac{1}{r} \left(\frac{\partial u_r}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} - u_{(\theta+\tau)\mathrm{mod}(2\pi)} \right) \right) v_r \\
&+ \left(u_x \frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial x} + u_r \frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial r} \right. \\
&\left. + u_{(\theta+\tau)\mathrm{mod}(2\pi)} \frac{1}{r} \left(\frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} + u_r \right) \right) v_{(\theta+\tau)\mathrm{mod}(2\pi)} \\
&- p \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{1}{r} \frac{\partial v_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} + \frac{v_r}{r} \right) \\
&+ \frac{1}{Re} \left(\frac{\partial u_x}{\partial x} \frac{\partial v_x}{\partial x} + \frac{\partial u_x}{\partial r} \frac{\partial v_x}{\partial r} + \frac{1}{r} \frac{\partial u_x}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} \frac{1}{r} \frac{\partial v_x}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} \right. \\
&+ \frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial x} + \frac{\partial u_r}{\partial r} \frac{\partial v_r}{\partial r} \\
&+ \frac{1}{r} \left(\frac{\partial u_r}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} - u_{(\theta+\tau)\mathrm{mod}(2\pi)} \right) \frac{1}{r} \left(\frac{\partial v_r}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} - v_{(\theta+\tau)\mathrm{mod}(2\pi)} \right) \\
&+ \frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial x} \frac{\partial v_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial x} + \frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial r} \frac{\partial v_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial r} \\
&\left. + \frac{1}{r} \left(\frac{\partial u_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} + u_r \right) \frac{1}{r} \left(\frac{\partial v_{(\theta+\tau)\mathrm{mod}(2\pi)}}{\partial (\theta+\tau)\mathrm{mod}(2\pi)} + v_r \right) \right).
\end{aligned}
$$

Therefore Equation (85) is O(2)-equivariant, for symmetrical inlet conditions (the Poiseuille flow). It is however not O(2)-equivariant for θ-dependent inlet flow.