

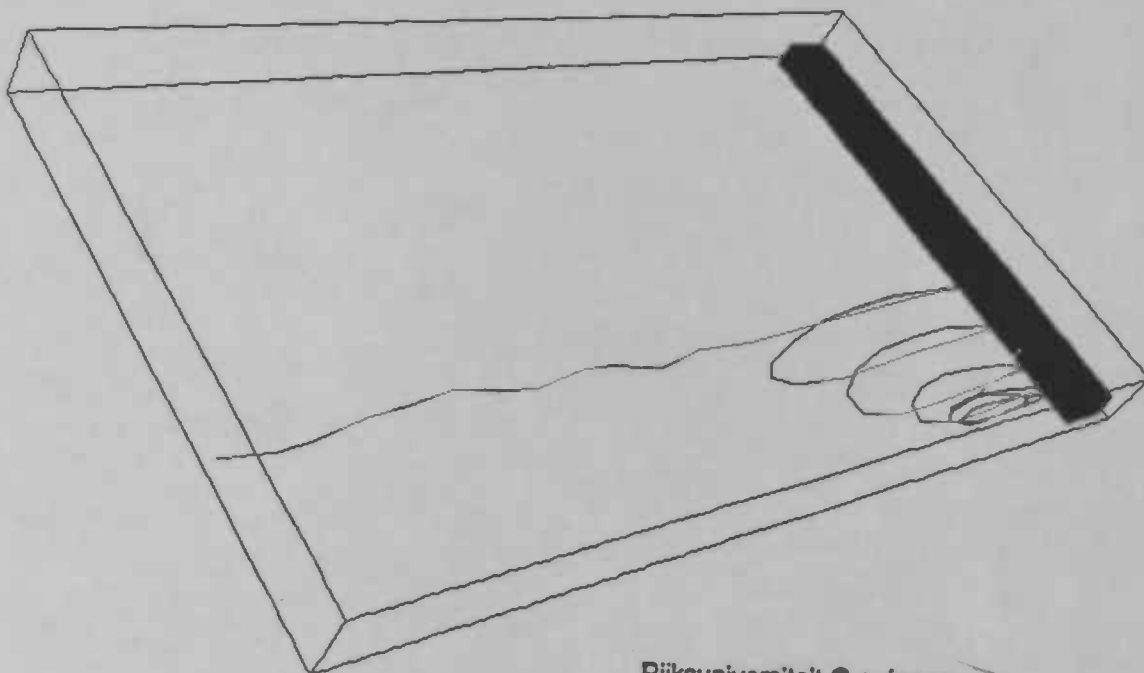
WORDT
NIET UITGELEEND

Bieb 1



Numerical Simulation of Channel Flow with Separated Flow Regions

Bernard Bos



Rijksuniversiteit Groningen
Bibliotheek
Wiskunde / Informatica / Rekencentrum
Landleven 5
Postbus 800
9700 AV Groningen

Department of
Mathematics

RuG



Master's thesis

Numerical Simulation of Channel Flow with Separated Flow Regions

Bernard Bos

Rijksuniversiteit Groningen
Bibliotheek
Wiskunde / Informatica / Rekencentrum
Landleven 5
Postbus 800
9700 AV Groningen

University of Groningen
Department of Mathematics
P.O. Box 800
9700 AV Groningen

August 1998

Preface

During the last year I have been working on this thesis. The motivation was based on a practical problem brought forward by the Gasunie.

I would like to thank Professor Veldman for his supervision, Jos Dijkstra for explaining his changes in ComFlo and Jeroen Gerrits for helping me with the CRAY. Finally I would like to thank all my friends and family for their support.

Contents

Preface	1
1 Introduction	3
2 Mathematical Model	5
2.1 Navier-Stokes Equations	5
2.2 Boundary Conditions	6
3 Numerical Model	7
3.1 Apertures	7
3.2 Labels	7
3.3 Discretised Navier-Stokes Equations	8
3.3.1 Discretisation in Time	9
3.3.2 Discretisation in Space	10
3.4 Boundary Velocities	12
3.4.1 BB- and BF-velocities	12
3.4.2 Velocities at the Inflow and Outflow	12
3.5 Solution Method	14
3.6 2D Problems	15
4 Results	16
4.1 2D Flow over a Backward Facing Step at $Re = 150$	17
4.2 Flow over a Backward Facing Step at $Re = 800$	20
4.2.1 2D Case	20
4.2.2 3D Case	22
4.3 Influence of a Welched Ribblet on Pressure Drops	24
5 Conclusions	28
A Program Description	29
A.1 Calling Sequence	29
A.2 Common Block Variables	30
A.3 Subroutines	33
A.4 Input and Output	34

Chapter 1

Introduction

We all know that in applied science models are being used. These models are usually mathematically formulated. This is also the case with models used in the area of fluid mechanics. In the first half of the 19th century two men, Navier and Stokes, presented a couple of equations which describe the evolution in time of a fluid flow. These equations have normally no analytic solution, except in a few simple cases. Nowadays we can solve these equations with numerical methods and with the help of powerful computers. We have to remember that when we use numerical methods, we will not get an analytic solution but just an estimate of the solution. This point has to be made because there are fluid flows which cannot be "solved" by any computer. In those cases the computer would take such a long time calculating that it would not be worth waiting for. Sometimes it is even impossible.

When there is no analytic solution of the Navier-Stokes equations we have to use numerical methods, requiring a computational grid. If this grid has not got enough grid points then the solution calculated by the computer will not be an estimate of the "real" solution, for example with turbulent flows. Unfortunately, in reality flows are turbulent. A company as the "Gasunie" is dealing with such flows in gas carriers.

There are a few points that have to be made clear about flows we encounter in reality. First, in reality flows are three-dimensional. However we try to model them two-dimensional because these problems are easier to solve by the computer. Second, to simulate turbulent flows we need very fine grids. Calculation on fine grids takes a lot of time and computing power.

We have now explained a few problems of the flows we encounter in reality. These turbulent flows and their study, however are of great importance, therefore there are models that only describe these turbulent flows. These models use less computing power then the full Navier-Stokes equations. A problem with these models is that they contain parameters which have to be tuned for each separate problem.

In this thesis we will use a Direct Numerical Simulation (DNS)-solver ComFlo by Gerrits [4]. This DNS-solver uses a Cartesian grid approach. One of the ingredients of this DNS-solver is the boundary handling. In the thesis by Dijkstra [3] a more accurate boundary handling has been developed. This thesis gives also a small report of an industrial problem put forward by the "Gasunie". Our goal is to investigate the influence of the rough innersurface of gas pipes on pressure drops. The first simulations have been done by Dijkstra and the results are discussed at the end of his thesis. These results where not satisfying. So Dijkstra came to the conclusion that there has to be done some further investigation for this project. One of the areas that needed some improvement was the numerical stability of ComFlo.

To improve the numerical stability of ComFlo we have changed the program in three areas, namely:

- the in- and outflow conditions
- the buffer zone
- the time integration

In Chapter 3 we will explain the changes made in ComFlo in more detail. In Chapter 4 we will evaluate the current potential of ComFlo as a simulation method for turbulent flows. It describes in the first two sections the results of a laminar flow over a backward facing step. We start with the 2D problems at $Re = 150$. In the next section we simulate the flow at $Re = 800$ as a 2D and 3D problem. We will show in that section that $Re = 800$ cannot be modelled as a 2D flow. The final section contains a simulation of a turbulent flow in a gas pipe. In Chapter 5 conclusions will be made regarding our results and that also ends our thesis.

Chapter 2

Mathematical Model

2.1 Navier-Stokes Equations

In this thesis we study the flow of an incompressible fluid in a 3D complex domain Ω . The equations which describe the evolution in time of such a flow are the Navier-Stokes equations

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (2.1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right), \quad (2.2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right), \quad (2.3)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right). \quad (2.4)$$

In these equations ρ is the density, which in our case (incompressible fluid) is constant. The kinematic viscosity is ν , which is equal to $\frac{\mu}{\rho}$, where μ is the dynamic viscosity. Furthermore p stands for the pressure, and the variables u , v and w are the velocities in respectively the x -, y - and z -direction. Equation (2.1) expresses the conservation of mass and the other equations (2.2), (2.3) and (2.4) express the conservation of momentum in respectively the x -, y - and z -direction. In the case that the fluid is compressible we refer to e.g. [6].

We can also write these equations in vector notation with $u = (u, v, w)^T$:

$$\text{div } u = 0, \quad (2.5)$$

$$\frac{\partial u}{\partial t} + (u \cdot \text{grad})u = -\frac{1}{\rho} \text{grad } p + \nu \text{div grad } u. \quad (2.6)$$

Furthermore we need the equations in conservation form. This will be necessary, because in the numerical model the equations are discretised using conservation cells. If these changes are made then the equations can be written as:

$$\text{div } u = 0, \quad (2.7)$$

$$\frac{\partial u}{\partial t} + \text{div}(uu^T) = -\frac{1}{\rho} \text{grad } p + \nu \text{div grad } u. \quad (2.8)$$

The equations (2.7) and (2.8) can also be made dimensionless by scaling them with a characteristic length L and a characteristic velocity U . We now get:

$$\operatorname{div} \mathbf{u} = 0, \quad (2.9)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\mathbf{u}\mathbf{u}^T) = -\operatorname{grad} p + \frac{1}{Re} \operatorname{div} \operatorname{grad} \mathbf{u}. \quad (2.10)$$

We have now reduced the Navier-Stokes equations to the equations (2.9) and (2.10) where just one parameter is of importance, namely the Reynolds number Re . The Reynolds number tells us if the convective terms dominate the diffusive terms or otherwise. The Reynolds number is defined as:

$$Re = \frac{UL}{\nu}. \quad (2.11)$$

2.2 Boundary Conditions

We will now handle the boundary conditions which are needed to solve the partial differential equations. The domain Ω has a boundary $\partial\Omega$. We assume that a part of $\partial\Omega$ is not a wall, say Γ . First we will handle two possible conditions on the wall namely free-slip and no-slip. Both conditions express the impenetrability of the wall, which is that the normal part of the velocity is zero at the wall. The difference in both conditions is the tangential part of the stress at the wall. If we are dealing with a viscous fluid near the wall, then we use the no-slip condition, which tells us that the tangential stress does not vanish at the wall. In the other case we assume that the stress vanishes. If we formulate this mathematically then we get:

$$\mathbf{u} = 0 \text{ on no-slip part of } \partial\Omega \setminus \Gamma \quad (2.12)$$

$$u_n = 0, \frac{\partial}{\partial n} u_t = 0 \text{ on free-slip part of } \partial\Omega \setminus \Gamma \quad (2.13)$$

In this report we will always use driven flows, for these flows we need an inflow and an outflow condition. For the part Γ_{in} of Γ , the inflow, we use a Dirichlet condition. The other part called Γ_{out} is used for the outflow. For the boundary condition at the outflow there are a lot of possibilities, for more detail see [3]. We have used a boundary condition that makes use of the following points, which all have a physical interpretation:

1. The conservation of flux: $\operatorname{flux}(\Gamma_{in}) = \operatorname{flux}(\Gamma_{out})$
2. $\frac{\partial p}{\partial n} = \text{Constant}$ at Γ_{out}
3. In the tangential direction a free-slip condition: $\frac{\partial}{\partial n} u_t = 0$
4. The conservation of mass using (3): $\frac{\partial^2 u_n}{\partial n^2} = 0$

The physical interpretation is that the points 1 and 4 are both directly related to the physical law: conservation of mass. Point 2 tells us something about the pressure drop in a pipe, which is constant if there are no obstacles. Point 3 is already explained. The mathematical model has now been completed. Later on we will see that the above boundary conditions fit perfectly in the numerical model, which makes use of conservation cells.

Chapter 3

Numerical Model

The mathematical model presented in the previous chapter was based on a complex shaped domain Ω in 3D. For ease of presentation we will describe the numerical model for a domain in 2D, although extension to 3D is straightforward.

3.1 Apertures

The numerical model uses a Cartesian grid. The problem of Cartesian grids is that it is possible that a curved boundary can cross a grid line at a place which is not a grid point. Therefore we introduce apertures, which take values between 0 and 1. There are two different kinds of apertures; the exact use of these apertures is described in [4].

1. Volume apertures F_b : is placed in the middle of each cell, indicating the amount of fluid in a cell.
2. Edge apertures A_x, A_y, A_z : is placed in the middle of a cell face, indicating the fraction of the cell face open for flow.

One of the purposes of apertures is labeling, to be explained in the following section.

3.2 Labels

In this numerical model we give each cell a label. The labels give an indication of the function of each cell. For this labeling we use the Volume apertures F_b . If $F_b \geq \frac{1}{2}$ then the volume of the cell is filled with more than 50% fluid and is called an F (luid)-cell. If $F_b < \frac{1}{2}$ then the cell is called a B (oundary)-cell if at least one neighbor is an F -cell. Cells are neighbors if they have at least one common cell face. In 2D four neighbors is the maximum for a cell, in 3D this is six. The cells that remain are called X (terior)-cells and are of no importance for the numerical model. In this thesis we will always simulate a driven flow, we therefore have the possibility to change some B -cells into I -cells and some F -cells into O -cells. An I -cell will represent the inflow and an O -cell will represent the outflow.

Now we will place the variables p , u , v and w . This model uses a staggered grid. When we use a staggered grid, we place the pressure in the middle of each cell and the velocities in the

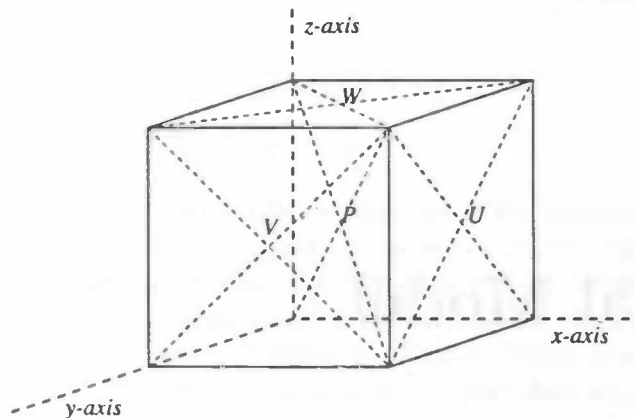


Figure 3.1: The places of the variables u , v , w and p on a grid for a domain in 3D.

middle of each cell face, see figure 3.1. We will place the unknown pressure in the F -cells, the pressure in the other cells will be eliminated or given a value, see [3]. We have now labeled the cells and will use these labels for labeling the velocities between combinations of the different cells. We have the following possibilities:

- FF, FO -velocities will be computed from the momentum equation
- BF, BB -velocities will be computed from the boundary condition
- FI -velocities will be given a Dirichlet condition
- BO -velocities will be given an appropriate value, but not a Dirichlet condition, see section 3.4.2
- II -velocities will be given a Dirichlet condition
- FX, OX -velocities cannot appear because there has to be a B -cell between an F, O -cell and an X -cell
- BX, IX -velocities are not important, because these velocities are not in the discretised domain

The simple grid in figure 3.2 gives an example of all possible labels. The circles mark the places of the velocities.

3.3 Discretised Navier-Stokes Equations

We already discussed the apertures and the labeling. These tools are needed when we want to solve the Navier-Stokes equations with our numerical model. But first we have to discretise the equations in time and space.

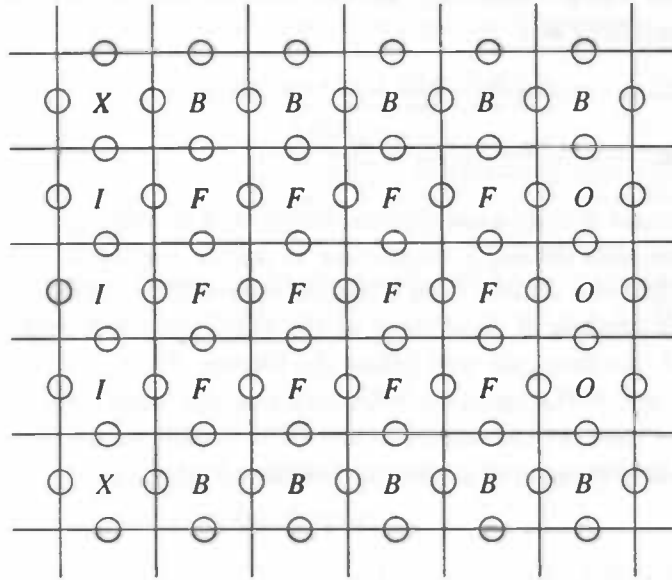


Figure 3.2: A possible grid for a 2D channel flow, with cell names

3.3.1 Discretisation in Time

The Navier-Stokes equations (2.7) and (2.8) are discretised in time explicitly. If we use the *Forward-Euler* method, a simple explicit integration method, then we get:

$$\operatorname{div} u^{n+1} = 0, \quad (3.1)$$

$$\frac{u^{n+1} - u^n}{\delta t} + \operatorname{grad} p^{n+1} = R^n, \quad (3.2)$$

where R^n is defined as:

$$R^n = -\operatorname{div} (u^n u^{nT}) + \nu \operatorname{div} \operatorname{grad} u^n. \quad (3.3)$$

One of our improvements is a better integration method in time. The method we have chosen is the *Adams-Bashforth* method. The *Adams-Bashforth* method is $O(\delta t^2)$ in contrary to the *Forward-Euler* method, which is $O(\delta t)$. The new integration method makes it possible to choose a larger time step δt . If we use this method to integrate the equations in time then (3.2) will change and we get:

$$\operatorname{div} u^{n+1} = 0, \quad (3.4)$$

$$\frac{u^{n+1} - u^n}{\delta t} + \operatorname{grad} p^{n+1} = \frac{3}{2}R^n - \frac{1}{2}R^{n-1}. \quad (3.5)$$

We have given the velocities a superscript n which indicates at what time level the velocity field has been calculated, so u^n stands for the velocity field at $t_n = n\delta t$. We have discretised the pressure at the new time level $(n+1)\delta t$ to make sure that the new velocity field is divergence free.

Although we can choose a larger time step, we have to fulfill two conditions to assure numerical stability. The two conditions are:

1. The CFL-condition : $\max \left\{ \frac{|u|\delta t}{\Delta x}, \frac{|v|\delta t}{\Delta y}, \frac{|w|\delta t}{\Delta z} \right\} \equiv \text{CFL} \leq 1$
2. The D-condition : $\frac{2\nu\delta t}{(\min \{\Delta x, \Delta y, \Delta z\})^2} \equiv D \leq 1$

The CFL-number changes during a simulation, because of the changing velocities. Therefore we will halve the time step during a simulation to assure numerical stability. If the CFL-number gets higher than the limiter then the time step will be halved. We double the time step to keep the total number of time steps of the simulation minimal. The time step will be doubled if the CFL-number gets well below the limiter. Note: The time step will only be doubled when the D- and CFL-condition hold with the new time step.

During simulations we have set the limiter of the CFL-condition to 0.40 or lower. In practice a lower value for the limiter assures numerical stability better.

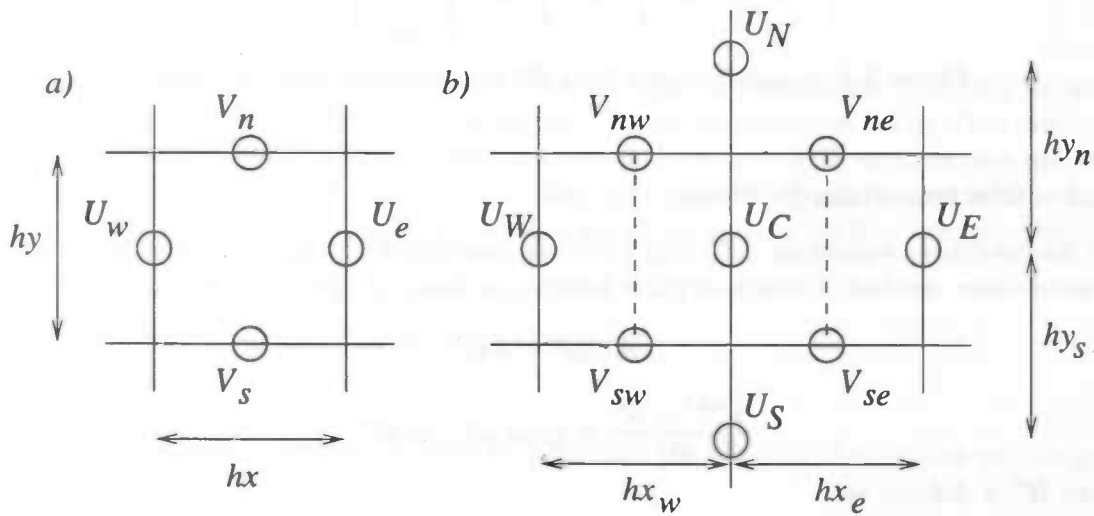


Figure 3.3: The conservation cells for respectively the equations (3.4) and (3.5)

3.3.2 Discretisation in Space

We will now discuss the discretisation in space, how this exactly works is handled in [3]. The bottom line is that velocities near the boundary are discretised using a modified scheme. The *FF*, *FO*-velocities are discretised normally. First we will discretise (3.4). To discretise the continuity equation we use the conservation cell that is given in figure 3.3(a) and get the following:

$$\text{div } u^{n+1} \doteq \left(\frac{v_e^{n+1} - v_w^{n+1}}{hx} + \frac{v_n^{n+1} - v_s^{n+1}}{hy} \right). \quad (3.6)$$

We still have one equation left to discretise, this will be done also with a conservation cell. For this momentum equation we use the cell in figure 3.3(b) with as corners the V -velocities.

The momentum equation can be split into two parts: the convective terms and the diffusive terms. First we discretise the convective terms and get the following:

$$\left(\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right) \Big|_C \doteq \frac{\frac{1}{4}((u_E + u_C)^2 - (u_C + u_W)^2)}{\frac{1}{2}(hx_w + hx_e)} + \frac{\frac{1}{4}((u_N + u_C)(v_{ne} + v_{nw}) - (u_C + u_S)(v_{se} + v_{sw}))}{\frac{1}{2}(hy_n + hy_s)}, \quad (3.7)$$

Now we will discretise the diffusive terms, beginning with the gradient of the pressure. We know the pressures are placed in the middle of each cell, so we get for the first component of the gradient:

$$\frac{\partial p^{n+1}}{\partial x} \Big|_C \doteq \frac{p_e^{n+1} - p_w^{n+1}}{\frac{1}{2}(hx_w + hx_e)} = G_h p^{n+1}, \quad (3.8)$$

where G_h is the discretised grad operator that later on will be used in section (3.5). The (div grad) operator will now be discretised and we get:

$$\begin{aligned} \text{div grad } u^n \Big|_C &\doteq \frac{hx_e u_W^n - (hx_w + hx_e)u_C^n + hx_w u_E^n}{\frac{1}{2}(hx_w + hx_e)hx_w hx_e} \\ &+ \frac{hy_s u_N^n - (hy_n + hy_s)u_C^n + hy_n u_S^n}{\frac{1}{2}(hy_n + hy_s)hy_n hy_s}. \end{aligned} \quad (3.9)$$

The Buffer Zone

When we are simulating flows with a high Reynolds number then we need some improvements of the numerical stability of the model. We have already discussed one of the three improvements made in ComFlo. The second one is the buffer zone. One of the problems of the numerical stability is caused by the outflow condition. We will explain these changes in the following section. To let the new outflow condition adapt more easily to the pipe or channel flows, we have implemented a buffer zone. In this buffer zone we decrease the Reynolds number so that at the end of the pipe or channel the flow will be laminar. One of the parameters where the Reynolds number is based on is the kinematic viscosity ν . In the buffer zone we have implemented a function $\nu(x, y, z)$, which increases ν at the end of the pipe. In (2.2), (2.3) and (2.4) we see that ν is placed in front of the (div grad) operator. This is correct when ν is constant, but when we want to increase the viscosity at the end of the pipe slightly then the viscosity depends on the variables (x, y, z) . The viscosity is placed between the div and grad operator, see [6]. So when we change only the viscosity for example in the x -direction then we get instead of (3.9) the following:

$$\begin{aligned} \text{div } (\nu \text{ grad } u^n) \Big|_C &\doteq \frac{\nu_w hx_e u_W^n - (\nu_e hx_w + \nu_w hx_e)u_C^n + \nu_e hx_w u_E^n}{\frac{1}{2}(hx_w + hx_e)hx_w hx_e} \\ &+ \frac{\nu_n hy_s u_N^n - (\nu_s hy_n + \nu_n hy_s)u_C^n + \nu_s hy_n u_S^n}{\frac{1}{2}(hy_n + hy_s)hy_n hy_s}, \end{aligned} \quad (3.10)$$

where for example ν_w is the weighted average of the viscosities ν_C, ν_W . The positions of the viscosities, for example ν_w, ν_C are the positions of the velocities u_W, u_C . We have to remark here that the viscosities ν_n and ν_s are the same.

Higher Order

If we look at the discretisation in space, we see that the discretisation is a 2nd order method. So the leading term of the truncation error is $O(\Delta x^2)$. If we eliminate this leading term in the truncation error then we get a 4th order method. We get a 4th order method, because in the truncation error there is no $O(\Delta x^3)$ term. For this elimination we use a method that is equivalent to the Richardson extrapolation and can be split up into 3 phases.

First, we choose a new control volume for the momentum equations. This second volume is 3 times larger than the small control volume. Second, we discretise the momentum equations with the 2nd order method using the new control volume. We have a truncation error that is exactly 9 times larger than the one in case of the small control volume. Third, we can now subtract $1/8$ of the new volume from $9/8$ of the small volume. We have eliminated the 2nd order terms and have a 4th order method, for more detail see [11].

3.4 Boundary Velocities

3.4.1 BB- and BF-velocities

Boundary velocities are needed when we are discretizing near boundaries. Normally we would use virtual velocities for the discretisation scheme. In our numerical model another method is implemented, which is more effective than the standard method. This method uses the exact boundary condition. The *BB*- and *BF*-velocities are computed by extra- and interpolation, this depends on whether the boundary velocities are in- or outside the flow domain. For more see [4].

3.4.2 Velocities at the Inflow and Outflow

Inflow

At the inflow we have a Dirichlet condition. Therefore the *FI*- and *II*-velocities will be given a value.

Outflow

We have seen in the previous chapter that our treatment of the outflow is based on four points. We have already shown that the outflow condition has a physical interpretation. With the help of figure 3.4 we will now show how the boundary velocities (in our example) U_c, U_e, V_{ne}, V_{se} are computed.

We start with equation (3.11), which is one of the equations that will come forward in the construction of the *Pressure Poisson*-equation, to compute the *FO*-velocity U_c .

$$\frac{U_c^{n+1} - U_c^n}{\delta t} = -G_h p^{n+1} + R^n \quad (3.11)$$

We assume that $-G_h p^{n+1} = \text{Constant}$ so if we substitute this in (3.11), we get:

$$\frac{U_c^{n+1} - U_c^n}{\delta t} = \text{Constant} + R^n \quad (3.12)$$

We can now use the conservation of flux and calculate both of the fluxes at the inflow and outflow. The flux at Γ_{in} will remain constant during the whole calculation because here a Dirichlet condition is being used.

$$Inflow = \int_{\Gamma_{in}} U d\Gamma_{in} \quad (3.13)$$

$$Outflow = \int_{\Gamma_{out}} U_c d\Gamma_{out} \quad (3.14)$$

We have to remark here that Γ_{out} is the boundary where the U_c -velocities are defined. In the first time step we will choose a boundary condition so that the *Inflow* will be equal to the *Outflow*. Our first outflow condition will be in our example:

$$U_e = \frac{Inflow}{\int_{\Gamma_{out}} d\Gamma_{out}} \quad (3.15)$$

So we begin with a uniform velocity field at the outflow in the first time step. From now on we will adapt this velocity field in each time step. In our example this will be U_e by calculating the *Outflow* with the U_c -velocities.

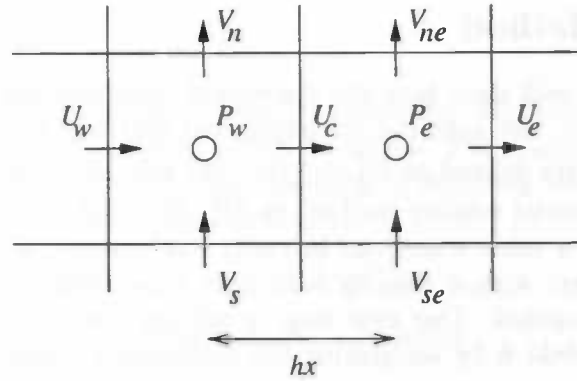


Figure 3.4: Two cells in the x -direction with at the left an F -cell and at the right a O -cell.

We now have a new velocity field for the whole domain such that a new *Outflow* can be calculated. If we substitute the definitions for *Inflow* and *Outflow* in equation (3.12) knowing that the "old" *Outflow* is equal to the *Inflow* we get:

$$\int_{\Gamma_{out}} U_c^{n+1} - U_c^n d\Gamma_{out} = \int_{\Gamma_{out}} \delta t [\text{Constant} + R^n] d\Gamma_{out} \quad (3.16)$$

$$Outflow - Inflow = \int_{\Gamma_{out}} \delta t [\text{Constant} + R^n] d\Gamma_{out} \quad (3.17)$$

$$0 = \int_{\Gamma_{out}} \delta t [\text{Constant} + R^n] d\Gamma_{out} \quad (3.18)$$

From (3.18) we can determine the new Constant. If we substitute this Constant in (3.12) then we obtain U_c^{n+1} .

Subsequently, to determine R^{n+1} on Γ_{out} and on the new time level U_e is required. Conservation of mass plays an important place in our numerical model and will be used here for the second time. We assume that there is conservation of mass over the two cells. We know that in the F - and O -cell this assumption is correct. We also know the boundary condition in the tangential direction, which tells us that the velocities V_{ne} and V_{se} respectively V_n and V_s must be equal. Therefore we set the velocities V_{ne}, V_{se} equal to V_n, V_s . If we do not use local refinements and put this information in the discretised equation for conservation of mass then we get:

$$U_w - 2U_c + U_e = 0 \iff \frac{\partial^2 u}{\partial n^2} = 0 \quad (3.19)$$

We can now evaluate the velocity U_e , our new velocity for the outflow.

$$U_e = 2(U_c + \delta t \text{ Constant}) - U_w \quad (3.20)$$

This concludes our boundary velocities at the outflow.

We have to remark here that because we are using a grid we have to approximate the integrals. This gives a numerical error in the calculation of the Constant.

3.5 Solution Method

In this final section we will show how the discretised equations are being solved. First we introduce some notation. We split the computational grid into two parts Ω_h^f and Ω_h^b . The FF - and FO -velocities are defined on Ω_h^f and the BF , BB , BO - and FI -velocities on Ω_h^b .

We will start with an initial velocity field u^n on Ω_h^f . If $n = 0$ then we set the velocity field equal to zero, or another value which can be easily implemented and is a good estimate. If $n \neq 0$ then we will start with a velocity field from a previous simulation. The velocities on Ω_h^b will now be computed. Our next step in solving the equations is the construction of a temporary vector field \tilde{u} by integrating the momentum equation without the pressure gradient in time

$$\tilde{u} := u^n - \delta t \{ D_h(u^n u^{nT}) - \nu D_h G_h u^n \} \text{ on } \Omega_h^f. \quad (3.21)$$

Here G_h and D_h are the discretised versions of respectively the gradient and divergence operator. We now just have to solve

$$D_h u^{n+1} = 0, \quad (3.22)$$

$$u^{n+1} + \delta t G_h p^{n+1} = \tilde{u}. \quad (3.23)$$

We could now substitute (3.23) in equation (3.22), however D_h works on $\Omega_h^f \cup \Omega_h^b$ and needs the pressures in the B -cells. Therefore we split D_h into two parts

$$D_h = D_h^f + D_h^b, \quad (3.24)$$

where D_h^f and D_h^b are defined on respectively Ω_h^f and Ω_h^b . We can now substitute these new operators in (3.22) and get:

$$D_h^f u^{n+1} = -D_h^b u^{n+1}, \quad (3.25)$$

$$u^{n+1} + \delta t G_h p^{n+1} = \tilde{u}. \quad (3.26)$$

$D_h^b u^{n+1}$ cannot be computed, because they are unknown. Therefore we use the velocity field of a previous time level, making a numerical error $O(\delta t)$. This error has already been introduced by the time integration. Now it remains to solve:

$$D_h^f u^{n+1} = -D_h^b u^n, \quad (3.27)$$

$$u^{n+1} + \delta t G_h p^{n+1} = \tilde{u}. \quad (3.28)$$

Now we can substitute (3.28) in (3.27) and get the *Pressure-Poisson* equation:

$$D_h^f G_h p^{n+1} = \frac{D_h^f \tilde{u} + D_h^b u^n}{\delta t}. \quad (3.29)$$

From the *Pressure-Poisson* equation we can compute a new pressure distribution. This new pressure distribution gives the velocity field on Ω_h^f at the new time level when we substitute this back in (3.28):

$$u^{n+1} = \tilde{u} - \delta t G_h p^{n+1}. \quad (3.30)$$

We can now compute the boundary velocities, and have the velocity field on the entire grid.

3.6 2D Problems

In this thesis we will simulate some flows that are modeled as 2D problems. Although ComFlo is based on the 3D version of the Navier-Stokes equations, it is possible to compute these 2D problems. To keep the benefits of 2D problems, such as requiring less computing power, the following options have been implemented in ComFlo:

1. one cell in the z -direction
2. free-slip conditions on u, v -velocities in the z -direction
3. $w=0$

If we remember the physical interpretation of the free-slip condition then it is clear that the three points just made, change the 3D flow in a 2D flow.

Chapter 4

Results

Our goal in this thesis is the study of a flow over a welched ribblet at a high Reynolds number. Before we started this time-consuming simulation we have done the following simulations, which are all backward facing steps:

1. 2D channel flow at $Re = 150$, $H/S = 2.0$
2. 2D channel flow at $Re = 150$, $H/S = 3.0$
3. 2D channel flow at $Re = 800$, $H/S = 2.0$
4. 3D channel flow at $Re = 800$, $H/S = 2.0$

We have started with flows over a backward facing step, because these flows and the one over a welched ribblet have some similarities. These similarities are that in both cases there is a separated flow region created behind the object. Such a separated flow region has certain characteristics that can be shown in velocity profiles and reattachment lengths, see figure 4.1. Later on we will see that our results agree on these points with the literature.

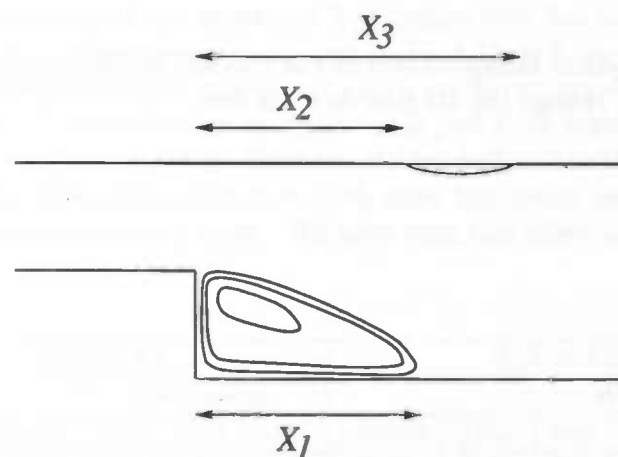


Figure 4.1: Reattachments lengths defined by Armaly [1]

4.1 2D Flow over a Backward Facing Step at $Re = 150$

To validate our implementations in ComFlo we will start with some simple test cases. We have done simulations on various uniform grids to verify the grid convergence. In the results only two grids will come forward, namely grid A (32×128) and grid B (64×256). Before we

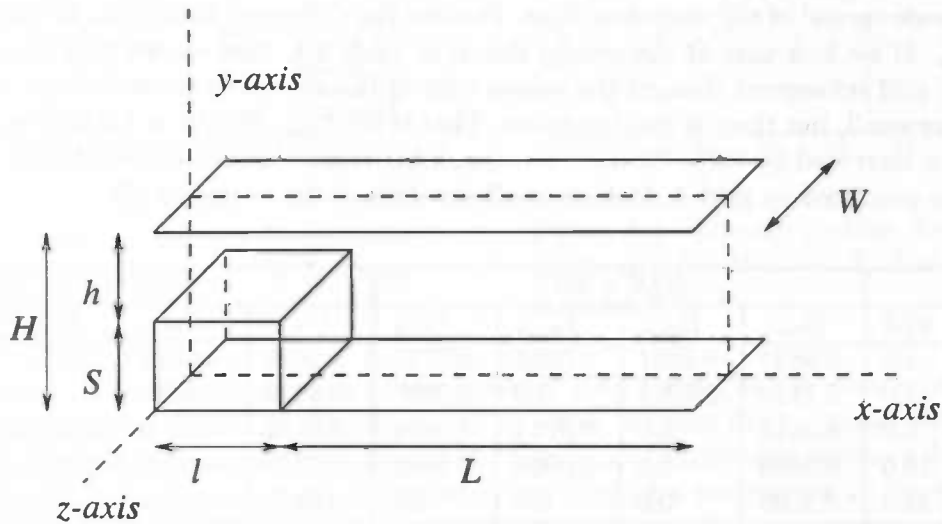


Figure 4.2: Geometry of the backward facing step

give the results, we give the starting conditions of the simulation. One of these conditions is the Reynolds number, which is based on a different length and velocity as in the $Re = 800$ cases, to be described in the next section. The Reynolds number here is defined as

$$Re = \frac{U_{max} S}{\nu}. \quad (4.1)$$

Also we have a parabolic velocity profile at the inflow with U_{max} scaled on 1. Furthermore we have $L = 19, l = 3, S = 0.5$ and $\nu = 3.33 \times 10^{-3}$ in both cases. We have also $x = 0$ at the end of the step. In both simulations we start with a time step of $\delta t = 0.01$ and allow an error 10^{-4} in the SOR-iteration of the *Poisson*-equation.

x/S	H/S=2.0						H/S=3.0					
	$U_{max} \times 10$			$U_{min} \times 10$			$U_{max} \times 10$			$U_{min} \times 10$		
	min	max	exp	min	max	exp	min	max	exp	min	max	exp
1.6	8.83	9.30	9.20	-2.88	-0.42	-0.91	9.19	9.79	9.72	-0.85	-0.57	-0.70
4.0	6.31	7.33	7.11	-0.66	-0.10	-0.16	8.60	9.28	9.28	-0.64	-0.21	-0.46
8.0	5.10	5.85	5.56	0.00	0.00	0.00	7.60	8.43	8.19	0.00	0.00	0.00

Table 4.1: Experimental data and the range of data produced by GAMM Workshop

The variable H is adjusted for each case. When the CFL-number gets too high then the time step δt is adjusted in the program. During our simulations these adjustments were not necessary, so during the simulation δt stayed the same.

First we will compare our results in table 4.2 with the data from the workshop in table 4.1. We see that the data lies in the wide range that is produced by the workshop. If we compare our data with the experimental data then we see that the agreement is very good in comparison with the wide spread of the workshop data. Further the difference with U_{max} is smaller than with U_{min} . If we look only at the results shown in table 4.2, then we see that at almost all places the grid refinement changes the values only in the second or third decimal. All these changes are small, but there is one exception. That is the U_{min} at $x/S = 4.0$ $H/S = 2.0$, here the value is increased by 70%. We have to remark here that the new value is better than the one that is produced by grid A, looking at all the data of the workshop [9].

Grid	x/S	$H/S = 2.0$				$H/S = 3.0$			
		U_{max}	U_{min}	V_{max}	V_{min}	U_{max}	U_{min}	V_{max}	V_{min}
coarse	1.6	0.8845	-0.0891	0.0056	-0.0715	0.9636	-0.0672	0.0037	-0.0550
	4.0	0.7139	-0.0284	0.0	-0.0785	0.9026	-0.0425	0.0	-0.0670
	8.0	0.5618	0.0	0.0074	-0.0024	0.8132	0.0	0.0001	-0.0273
	16.0	0.5089	0.0	0.0004	-0.0002	0.7228	0.0	0.0041	-0.0014
	24.0	0.5060	0.0	0.0	0.0	0.6874	0.0	0.0011	-0.0009
fine	1.6	0.9029	-0.1011	0.0104	-0.0701	0.9707	-0.0683	0.0059	-0.0508
	4.0	0.7245	-0.0485	0.0	-0.0910	0.9103	-0.0512	0.0	-0.0672
	8.0	0.5594	0.0	0.0098	0.0	0.8176	0.0	0.0	-0.0304
	16.0	0.5024	0.0	0.0005	-0.0001	0.7255	0.0	0.0042	-0.0014
	24.0	0.4990	0.0	0.0	0.0	0.6891	0.0	0.0012	-0.0008

Table 4.2: Extreme velocities at different places on the x -axis for $H/S = 2.0$ and $H/S = 3.0$.

An interesting characteristic of the separated flow region that has been created is the reattachment length ($= X_1$). If we compare our data (table 4.3) with the one from the workshop then we see that we predict a higher value for X_1 then in the experiment. If we use the refined grid for the case of $H/S = 3.0$ then we see that this is just 6.7% higher, which is acceptable. For the case of $H/S = 2.0$ this is 9.7%. Both results are well within the range covered by the workshop participants. Interesting is that in both cases the coarse mesh differs less from the experimental data than the refined mesh. We have seen this effect also at some data of the extreme velocities.

$H/S = 2.0$					$H/S = 3.0$				
		Workshop					Workshop		
coarse	fine	exp	min	max	coarse	fine	exp	min	max
4.80	4.94	4.5	3.70	5.8	6.20	6.34	6.0	5.05	7.25

Table 4.3: Reattachment lengths from both cases.

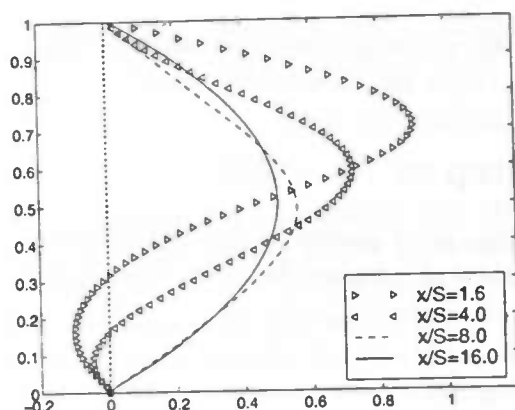


Figure 4.3: Velocity profiles downstream of the step for $H/S = 2.0$. The y -axis is scaled with H .

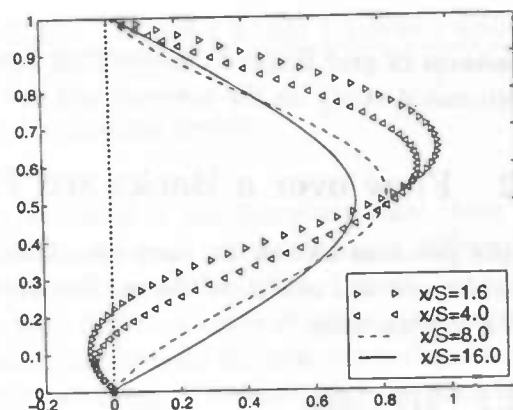


Figure 4.4: Velocity profiles downstream of the step for $H/S = 3.0$. The y -axis is scaled with H .

Finally we show some contour plots of the pressure distribution in the domain, see figure 4.5. In this figure we have the distribution of \bar{p} , which is equal to $(p - p_{step}) Re$. This is very interesting because we are interested in the influence of ribblets on pressure drops. The two contour plots in figure 4.5 are the same as the ones of a participant of the workshop. Therefore we have not included a contour plot of the workshop for a comparison.

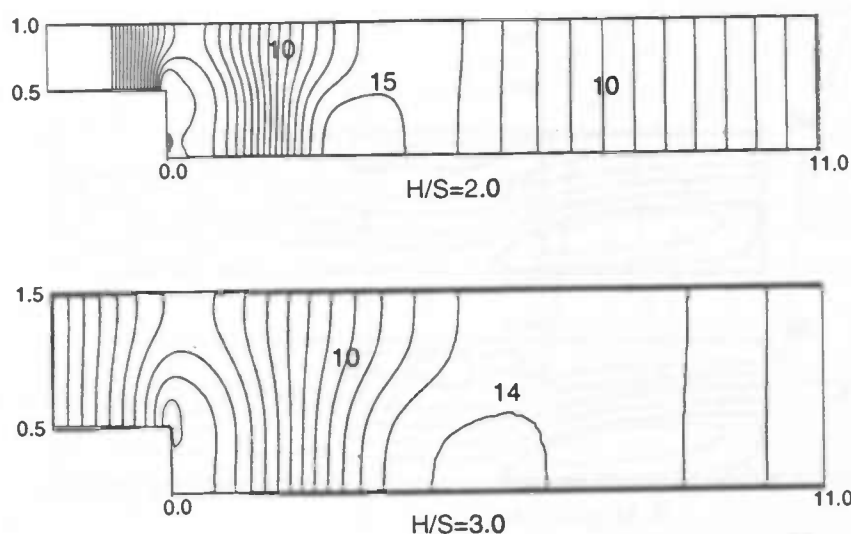


Figure 4.5: Contour plot of the pressure distribution \bar{p} in both cases.

After this discussion of the results, we can conclude that our implementations in the program are correct, because all our data lies in the range of the workshop and compare favorably with experiment. Generally, we can also say that the refinements have improved the results as we would have expected, except on the results for X_1 and some of the U_{min} at $H/S = 2.0$. A

refinement of grid B could confirm that this effect on X_1 was temporary. We have also done a refinement study on the pressure and the results here were the ones we expected.

4.2 Flow over a Backward Facing Step at $Re = 800$

In the previous section we have seen that ComFlo gives good results in the low Re cases. Therefore we will now show the results of some simulations at a higher Reynolds number, but still a laminar case.

4.2.1 2D Case

We start with a simulation that is interesting, because of the problems in the literature about the stability of this flow in question. The group of Princeton [7] says that the 2D flow is unstable. On the other hand there is a paper by Gresho *et al.* [5] that questions the method of the Princeton group. They show that the spectral-element method must be handled with care. It is not only the group of Princeton that says that the flow is unstable, but we compare our results with the ones we just have named.

The variables in this simulation are different from the one in the previous section. We have $L = 16, l = 1, S = 0.5, H = 1.0$ and $\nu = 1.25 \times 10^{-3}$. We have also a parabolic profile at the inflow with U_{max} scaled at 1.5. The Reynolds number is defined differently and is 800.

$$Re = \frac{2 U_{max} H}{3 \nu} = \frac{\bar{U} H}{\nu} \quad (4.2)$$

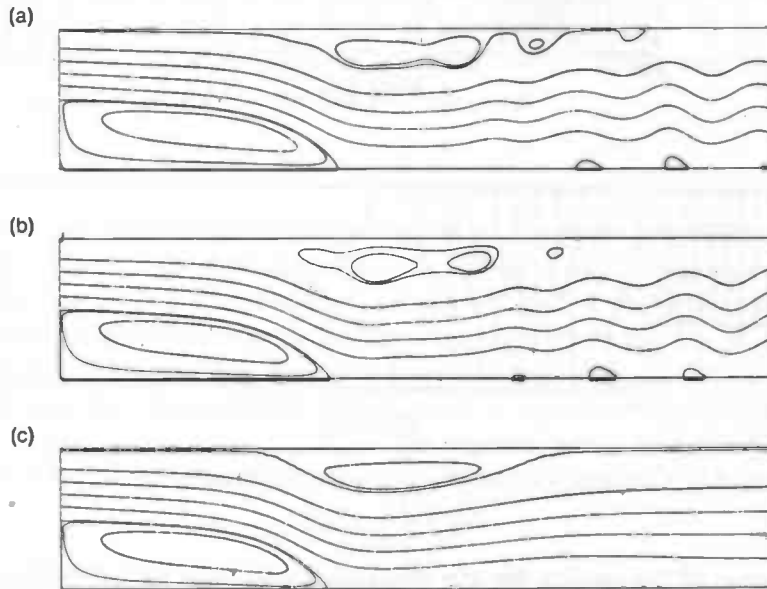


Figure 4.6: Streamlines for $Re = 800$: (a) Instantaneous flow on grid B (b) Instantaneous flow on grid A (c) Time-average flow on grid A

t	0.0000	0.5005	1.0010	1.5015	2.0020	2.5025	3.0030	3.5035	4.0040	4.5045
X_1	11.67	11.72	11.81	11.87	11.86	11.80	11.79	11.80	11.72	11.69

Table 4.4: Reattachment lengths over one period.

If we compare our parameters with the ones that are used in the literature then there is one difference, namely in the literature we have $l = 0$. This should not make a difference because the parabolic profile at the step should be the same as at the inflow, because we have a Poiseuille flow in this small pipe. We have also used different kinds of grids. We started with some coarse grids, because the simulations had to be run upto the non-dimensional time $t \approx 400$. The results of these coarse grids where not satisfying. Therefore we have used two grids that are also used by the time dependent finite difference method in [5], grid A (32×480) and grid B (64×960). The finite difference method uses a multi grid method including an even finer grid than grid B. But the amount of grid points of grid A and B should be enough, because our method includes a 4th order method contrary to the above described method, which is a 2nd order method.

The group of Princeton says that the 2D flow at $Re = 800$ locks into a two-frequency flow that influences the length of the first separated flow region. The quasi-periodic behavior of the flow should increase or decrease the reattachment length by 20%. Therefore we first looked at the effect on the length of the separated flow region. We can determine the reattachment length by taking the average of the data in table 4.4. The effect of this possible quasi-periodic behavior we measured was minimal and less than 1.0%.

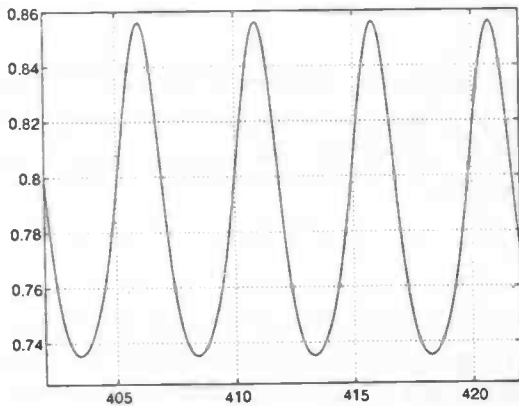


Figure 4.7: u reaching a periodic state of stability at P_1 .

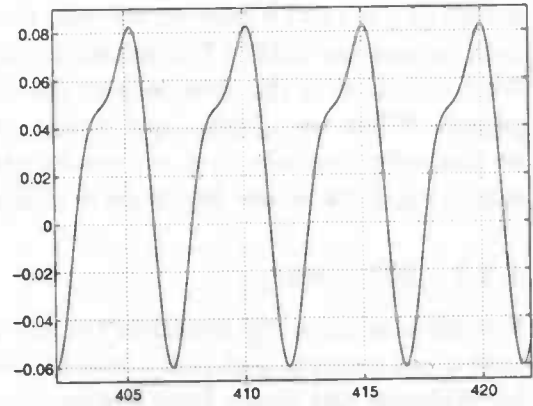


Figure 4.8: v reaching a periodic state of stability at P_1 .

Therefore we looked at the paper by Gresho *et al.* They say that the flow should become stable, unfortunately this is not confirmed by our simulation. In figure 4.6 it is clear that there are some vortices shed off near the wall. These vortices should damp out according to Gresho *et al.*, but our max norm of the velocity did not vanish during our simulation run but stayed constant. Some more investigation was required, we have therefore monitored some velocities at P_1 located at $(x = 10.0; y = 0.32)$ and P_2 located at $(x = 4.74; y = 0.41)$.

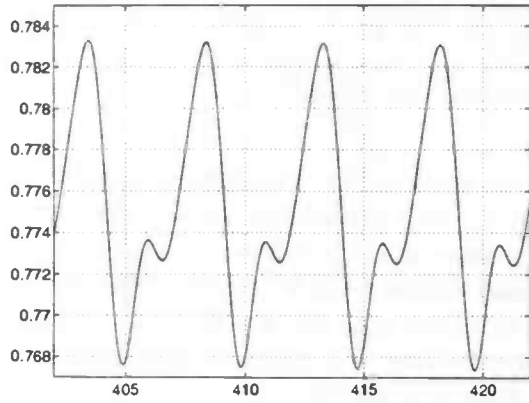


Figure 4.9: u reaching a periodic state of stability at P_2 .

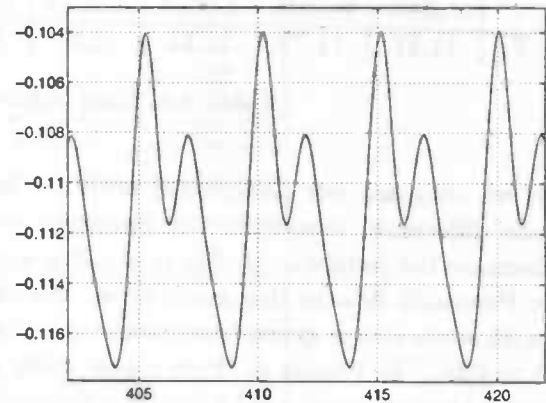


Figure 4.10: v reaching a periodic state of stability at P_2 .

In the figures 4.7 till 4.10 we see that the velocities have reached a steady oscillatory state with a main frequency $f_1 = 0.2050$. Also we see that the periodic behavior depends just on one main frequency and does not depend on two main frequencies. A periodic behavior can also be seen in the reattachment lengths see table 4.4. The small effect on the reattachment length could be explained because f_1 differs a factor 4 with one of the frequencies measured by the Princeton group, so the separated flow region cannot adapt quickly enough to the periodic behavior of the flow. If we compare the reattachment length averaged over one period ($X_1 = 11.77$), then we see that this is less than 4% lower than the ones obtained from both papers, see table 4.5 in subsection 4.2.2.

We conclude that the prediction of the reattachment length lies within the range of the two papers. When we compare our results of the stability of the flow with the two papers then we can only conclude that our results give no proof for one of the two papers. This is not surprising, because the literature is unclear on this point.

4.2.2 3D Case

We will now show the results of the 3D simulation at the same Reynolds number. We have used a non uniform grid ($132 \times 20 \times 24$) with Δx_{min} , Δy_{min} and Δz_{min} chosen as in the paper by Williams and Baker [12]. Furthermore we started with a uniform profile at the inflow, which was adjusted each time step by copying the velocity field of u of an intersection at $x \approx 0.6$ to the inflow. This can also be seen as periodic boundary condition in the small pipe. Furthermore we have for the geometry $S = 0.49$, $H = 1.01$, $L = 16$, $l = 1$ and $W = 18$. We adjusted the kinematic viscosity to the Reynolds number, which is defined as:

$$Re = \frac{2 U_{max} 2h}{3 \nu} \quad (4.3)$$

Interesting is that the Reynolds number is defined differently as in the 2D case but here the difference is very small. Because we used free-slip at the plane $z = 0$ our computational domain is $16.0 \times 1.01 \times 9.0$.

We are now able to compare the 3D simulation with the 2D simulation. The difference

becomes clear when we look at the reattachment length from both simulations in table 4.5. The prediction for X_1 of the 2D simulation is lower than the experiment of Armaly [1]. The prediction for X_1 of the 3D simulation is still lower than the experiment, but is well within a range of 6%. Furthermore our prediction is even better when we compare it with the data of Williams and Baker. Their prediction is also lower than the experiment.

2D	X_1	3D	X_1
ComFlo	11.77	ComFlo	13.47
Gresho <i>et al.</i>	12.20	Williams & Baker	13.90
Princeton group	12.25	Armaly <i>et al.</i>	14.20

Table 4.5: Reattachment length over one period.

The difference can be explained by the three-dimensionality of the flow. In figure 4.11, 4.12 and 4.13 we can clearly see the three-dimensionality.

First we will look at figure 4.11. When we look at this figure we see that near the no-slip wall (the top of the figure) there is a region where we have a "circulation" of the flow. In figure 4.12 we see a real circulation of the flow.

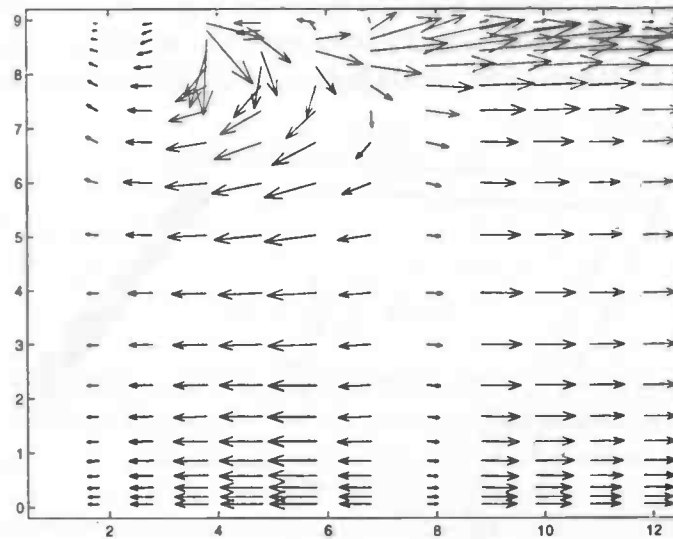


Figure 4.11: Velocity field near floor projected onto horizontal plane.

At the left-side of figure 4.11 we have the separated flow region as we expected. From this figure we can also determine the reattachment length X_1 by taking the length of the separated flow region at the bottom of the figure (the free-slip wall). Note: For determining X_1 we have to subtract the length of the step and scale it with S .

The second separated flow region can be seen in figure 4.12. It is near the free-slip wall where the velocities are plotted as very small arrows.

If we compare our results with the literature then we see that our data (see figure 4.11,

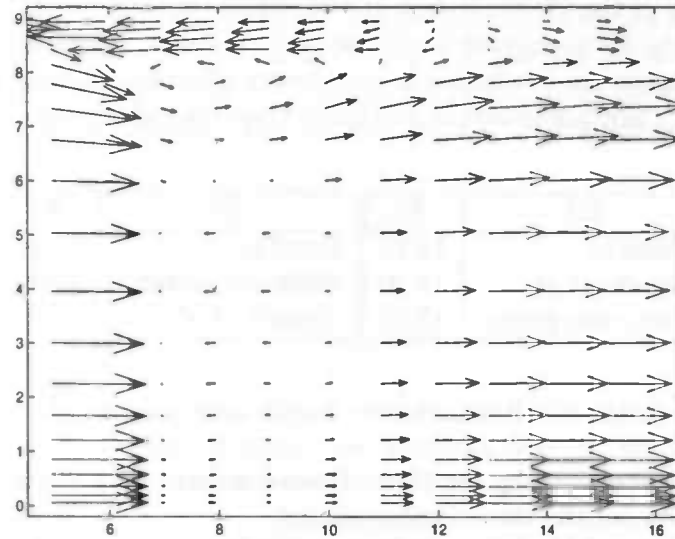


Figure 4.12: Velocity field near roof projected onto horizontal plane.

4.12 and 4.13) agree almost exactly with the figures in [12]. We have already seen that our reattachment length lies within the acceptable range of the experiment and the simulation of Williams and Baker. So therefore we conclude that in the 3D case we follow the literature.

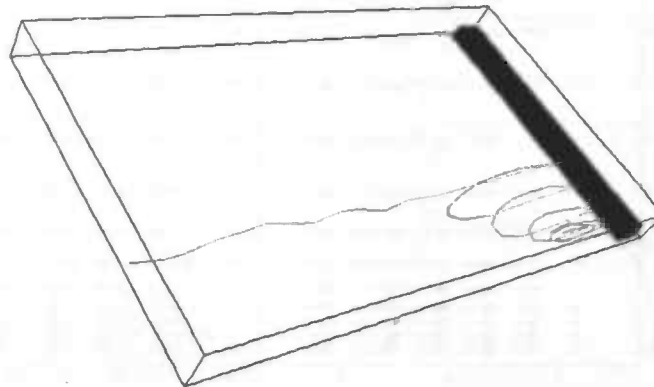


Figure 4.13: Streamline calculated from a point near the no-slip wall.

4.3 Influence of a Welched Ribblet on Pressure Drops

In this final section of this chapter we will show some results of a simulation of a channel flow with a welched ribblet, see figure 4.14. The original problem that was brought forward by the Gasunie was too difficult for our DNS-solver, because of the large Reynolds numbers involved.

Therefore we have done a simulation at a much lower Reynolds number ($Re_b \approx 5600$) than in the case of the Gasunie ($Re_b \approx 4 \times 10^6$). Although the Reynolds number is much lower, we are still dealing with a turbulent flow.

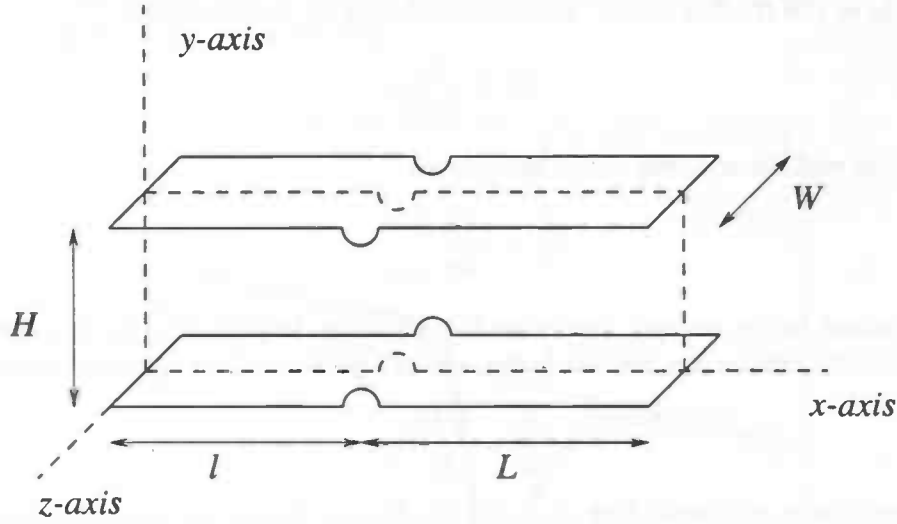


Figure 4.14: Geometry of the channel flow with a welched ribblet.

The Reynolds number is 5600 and is defined as:

$$Re_b = \frac{U_b H}{\nu}, \quad (4.4)$$

where U_b is the bulk velocity defined as:

$$U_b = \frac{1}{H} \int_0^H U dy. \quad (4.5)$$

The parameters of the geometry of this simulation are $H = 1, l = 6, L = 6$ and $W = 8$. We have chosen for the ribblet a radius of $\frac{1}{20}$ of H , which approximates the geometry conditions of a gas pipe, except for the total length. For a gas pipe we have that the distance between two ribblets is $120 \times H$. To reduce the high calculation times we assume that the effects of the ribblet on the flow are minimal at the end of the gas pipe. The kinematic viscosity is 1.8×10^{-4} and is increased in the buffer zone. We started with a uniform velocity profile at the inflow and with $\delta t = 2.0 \times 10^{-4}$. During the simulation our δt stayed in the interval $[1.0 \times 10^{-4}, 8.0 \times 10^{-4}]$.

We have only simulated the channel flow with a ribblet, because of the high calculation times. We have used non uniform grid ($128 \times 64 \times 40$), where the smallest cells are near the ribblet. The computations have been carried out on the CRAY J932 at the RUG. For the first 10 non-dimensional time units the program used 320 CPU hours. Kim *et al.* [8] used 250 CPU hours for a similar computation on the CRAY-XMP and they used a finer grid. They simulated a channel without a ribblet.

We will compare our results with data from a DNS-solver from the group of Kim. But also we

will compare the results with one of the experimental formulas from Dean [2]. Our comparison will be on several important characteristics. These characteristics are the friction factor C_f and the pressure drop dp/dx . To determine the effect of the ribblet on the pressure drop we will determine the dp/dx in front of and behind the ribblet from figure 4.17. First we give the definitions of the friction factor. The friction factor C_f is defined as:

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho U_b^2}, \quad (4.6)$$

where τ_w is the wall shear stress and is defined as:

$$\tau_w = \mu \left. \frac{\partial u}{\partial y} \right|_{y=0}. \quad (4.7)$$

From the friction factor we can determine the pressure drop dp/dx , because we have the following relation between the friction factor and the pressure drop for a channel flow:

$$\frac{dp}{dx} = \frac{\rho U_b^2}{H} C_f. \quad (4.8)$$

In the last relation p is independent of y . For the friction factor we have a formula by Dean based on experiments, which is very similar to the friction law for a pipe flow by Blasius. This formula is:

$$C_f = 0.073 Re_b^{-0.25}. \quad (4.9)$$

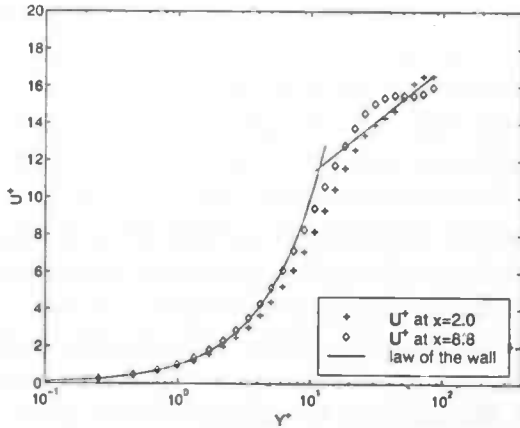


Figure 4.15: Velocity profile compared with the law of the wall.

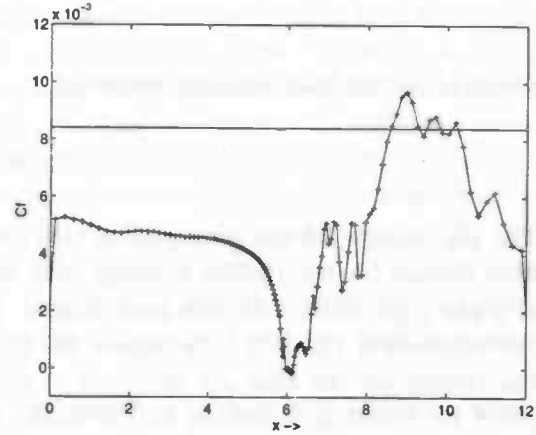


Figure 4.16: C_f versus x compared to Dean ($C_f = 0.0084$)

We will also check "the law of the wall" for this turbulent flow. In front of the ribblet ($x = 2.0$) we see that in figure 4.15 we follow the literature. If we look at the data from behind the ribblet ($x = 8.8$) then we see that we have a overshoot at the beginning of the inner layer. At the end of the inner layer we see that the data does not agree anymore. Important is the friction factor C_f , which depends on the data from the laminar sublayer. We see that both profiles follow the law of the wall here. In figure 4.16 we see that in front of the ribblet the

C_f is much lower than expected. The friction factor according to Dean and Kim *et al.* should be $C_f \approx 0.0084$. We have to remark here that Kim used almost two times more grid points in the y -direction. Also we see that just behind the ribblet the friction factor becomes negative. Behind the ribblet the friction factor increases and we see that the friction factor is higher than in front of the ribblet just as we expected. The dp/dx depends on the C_f , so we should get the same effects on the pressure drop.

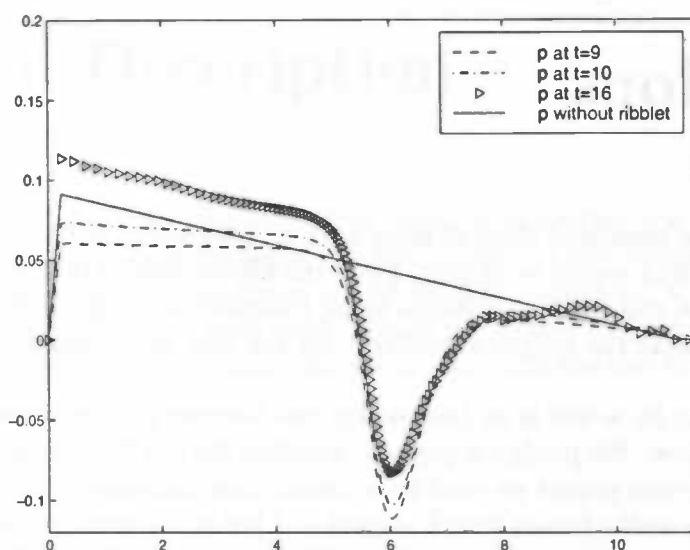


Figure 4.17: The time averaged pressure distribution at different time levels.

We have averaged the pressure in time to eliminate oscillations. In figure 4.17 we see that we have a constant pressure drop in front of and behind the ribblet before the simulation reached $t = 10$. Unfortunately we see that at $t = 16$ the pressure drop is not constant behind the ribblet. These disturbances in the pressure could be caused by possible vortices, which at that moment in time reach the end of the channel. These disturbances can be eliminated by running the simulation upto a much higher time level. We have just run the simulation upto $t = 20$.

We conclude that at the beginning of the simulation we have that the data follows our expectations. We have shown that the ribblet effects the pressure drop in a way as we expected when we compare both friction factors. Later on we see some disturbances in the flow and we cannot determine dp/dx anymore. For more accurate results we have to run the simulation to a much higher time level. Also we have to increase the amount of grid points. The paper by Kim, Moin and Moser used a non uniform grid ($192 \times 129 \times 160$). Although the domain is different than ours, they used almost two times more grid points in the y -direction.

Chapter 5

Conclusions

In this thesis we have simulated channel flows with an obstacle in the geometry. Behind this obstacle a separated flow region is created. From the simulations of the backward facing step at $Re = 150$ we have seen that our results agree favorably with the literature. From these results we conclude that the program ComFlo with the new implementations is an accurate DNS-solver.

The 2D simulation at $Re = 800$ is an interesting case because the literature is unclear about the stability of the flow. We predict a periodic stability for the 2D simulation. Our solution follows not one of the two papers we used for a comparison, but gives a new possible solution. The prediction of the reattachment length is good and lies in the range of the two papers. The results from the 3D simulations of the backward facing step at $Re = 800$ show clearly that this flow cannot be modeled as a 2D flow. These results agree very well with the literature. From the results of the $Re = 800$ case we can conclude that for a case with a high Reynolds number and an expected separated flow region, a 3D simulation is required. Therefore we have modeled the gas pipe as a 3D channel flow at $Re_b \approx 5600$. Although the Reynolds number is much lower than in case of a gas pipe ($Re_b \approx 4 \times 10^6$), the flow is turbulent. The results show clearly that for a more accurate investigation on the effect of the ribblet on dp/dx there is more time required. For example we have seen some disturbances in the pressure at the end of the channel. This is not surprising because we have just ran the simulation upto $t = 20$. For more accurate results we have to run the simulation upto a much higher time level. Another suggestion is to increase the amount of grid points. Unfortunately both suggestions will lead to even higher calculation times. The period of time for this project was too short to carry out such computations.

To improve the predictions of ComFlo for a turbulent channel flow with ribblet, there are some possibilities. For example we can use periodic boundary conditions at the inflow and outflow. Another possibility is to change the boundary conditions on the two side walls. Instead of taking for one side wall no-slip and for the other free slip, we can also use periodic conditions.

Appendix A

Program Description

In chapter 3 we have described the numerical model which solves the Navier-Stokes equations. Sometimes we have used the name ComFlo to refer to the numerical model. This is not correct, because ComFlo is the name of the computer program, a FORTRAN77-version where the numerical method has been implemented in. In this appendix we will discuss the calling sequence, the important variables, the subroutines and the files that ComFlo needs and produces.

A.1 Calling Sequence

The structure of the program comes forward in the calling sequence. The subroutines are called in the following order:

setup/pre-processing	SETPAR		
	GRID		
	LABEL	BNDDEF	
		BNDDEFIO	
time-integration	SETFLD/LOADFLD	BCIO	
		BC	
	COEFL		
	INIT		
	TILDE		
	SOLVEP	COEFR	
		PRESIT	SLAG
		BCIO	
		BC	
post-processing	AVS		
	MATLAB		
	SAVEFLD		

Later on we will explain shortly the subroutines. In the pre-processing the labels, apertures and velocity field are initialized. The velocity field is initialized by using a previous field from another simulation or by just giving it a starting value. Now we have the information to

calculate the left hand-side, i.e the matrix coefficients of the *Poisson*-equation. All these sub-routines are executed once in the pre-processing. We can now start with the time integration. The time integration will be repeated until the time limit is reached. Starting with the initialized velocity field we first calculate the vector field \tilde{u} . The subroutine SOLVEP will now calculate the pressure field. A new velocity field can be computed based on the new pressure distribution. Finally, boundary velocities are calculated from the new velocity field. At this stage all the temporal procedures have been executed and we can start with a new time-step. During the simulation there will be done some post-processing by AVS and MATLAB.

A.2 Common Block Variables

ADAM

UNN(I,J,K), VNN(I,J,K), WNN(I,J,K)

The velocities that are needed for the *Adams-Bashforth* integration method.

COEF1, COEF2 The coefficients that indicate which time integration method is used.

APERT

AX(I,J,K), AY(I,J,K), AZ(I,J,K)

The edge-aperture A_x between the cells $(i+1, j, k)$ and (i, j, k) .

The edge-aperture A_y between the cells $(i, j+1, k)$ and (i, j, k) .

The edge-aperture A_z between the cells $(i, j, k+1)$ and (i, j, k) .

FB(I,J,K)

The volume-aperture in cell (i,j,k) .

AVG

UAVG(I,J,K), VAVG(I,J,K), WAVG(I,J,K), PAVG(I,J,K)

The temporal averages of the velocities u , v , w and the pressure p

AVGSTART

The time level from which we start averaging the velocities and the pressure.

COEFP

DIV(I,J,K), CC(I,J,K), CXL(I,J,K), CXR(I,J,K), CYL(I,J,K),

CYR(I,J,K), CZL(I,J,K), CZR(I,J,K)

The coefficients that are used in the *Poisson*-equation.

DPDN

WE(I,J)

The velocities that are needed to calculate the *Outflow*.

EXACT

UEX(I,J,K), VEX(I,J,K), WEX(I,J,K)

The exact boundary conditions.

FLUID

NU

The kinematic viscosity.

GRIDAR

IMAX, JMAX, KMAX

The number of cells in each direction.

X(I), Y(J), K(Z)

The position of the grid lines.

DXP(I), DYP(J), DZP(K)

The distance between the grid lines in each direction, e.g. $DXP(I) = X(I) - X(I-1)$.

DXU(I), DYV(J), DZW(K)

The distance between the cell centers in each direction, e.g. $DXU(I) = 0.5 * (DXP(I) + DXP(I+1))$.

LABELS

ULABEL(I, J, K), VLABEL(I, J, K), WLABEL(I, J, K), PLABEL(I, J, K)

The velocity- and pressure-labels.

NEW

HIGH

Indicates if the 4th order method is used.

DIM

Indicates the dimension of the problem.

FS_3D

Indicates if there is a free-slip condition used on a particular wall, which at first was a no-slip wall.

NUMER

EPS

The maximum error in the pressure-iteration.

OMEGA, OMSTRT

The current and the starting relaxation parameter in SOR-iteration.

ITMAX, NOM

The total number of allowed iterations per time-cycle.

ITER

The number of iterations in current time-cycle.

C1, C2

The starting parameters of the time integration method.

PHYS

UN(I,J,K),VN(I,J,K),WN(I,J,K)

The velocities at the previous time-step.

U(I,J,K),V(I,J,K),W(I,J,K),P(I,J,K)

The velocities and pressure at the current time.

ROTATE

VOR,VORX,VORY,VORZ

The magnitude and the three components of the vorticity vector.

SMALL

HX,HY,HZ

The smallest cells in each direction, these values are used to calculate the CFL-number and also the D-number.

SPACE

DOMAIN

Indicates the domain of the problem.

XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX

Minimum and maximum of the domain in each direction.

STRGRD

XP,YP,ZP

The position of the concentration point.

ALPHAX,ALPHAY,APHAZ

The stretching parameters in each direction to stretch the grid.

TIME

TSTART,TMAX,T,DT,CYCLE

The starting time, the maximum simulation time, the current time, the current time step and the current time cycle.

ITERTOT

The total number of iterations during the simulation.

FRTOT

The total number of frames that will be produced from TSTART.

VAR

OBJECT

Indicates which object has been chosen.

LDFLD

Indicates if at the start of a new simulation, a previous field will be loaded.

VAR1, VAR2, VAR3, POSX, POSY, POSZ

The parameters that define the size and the position of the object.

FS, FSOBJ

Parameters that define the boundary conditions on the walls respectively object. The boundary condition can be free-slip or no-slip. Do not mistake them with FS_3D, because with this parameter we can make just one wall free-slip.

NRINTP

The number of integration points chosen to calculate the apertures.

A.3 Subroutines

We will now explain shortly the subroutines in alphabetical order:

AVS	generates data for the post-processing program AVS
BC	calculates the new boundary conditions and sets the exact boundary conditions
BCIO	sets the conditions for the velocities at the inflow and outflow
BNDDEF	defines the domain and objects and calculates the apertures
BNDDEFIO	defines the inflow and outflow boundaries
CELNR	defines in which cell the particle is
COEFL	calculates the coefficients for the left-hand side of the <i>Poisson</i> -equation
COEFR	calculates the coefficients for the right-hand side of the <i>Poisson</i> -equation
FREESLIP	changes the p-labels when the walls and/or object is free-slip
GRID	generates the stretched grid
INIT	initializes the velocity field for the new time-step
INTRPOL	interpolation of the velocities used in STREAM
LABEL	labels the velocities and pressure
LDFLD	loads the backup-file comflo-fld
MATLAB	generates data that can be used for the post-processing program MATLAB
PRESIT	calculates new pressure field
SAVEFLD	saves the velocity, pressure field to the backup-file comflo-fld Note: including the WE-velocities.
SETFLD	initializes the velocity and pressure field at $t = 0$
SETPAR	reads and sets the parameters that are given in the Input-file comflo-in
SOLVEP	calls PRESIT and calculates the new velocity field
STREAM	generates data that can be read by MATLAB to produce the trajectory of a particle in a 2D simulation
TILDE	calculates the temporary vector field

A.4 Input and Output

COMFLO-IN

In the previous section we have shown that in the subroutine SETPAR an input-file `comflo-in` is read. In this file the following parameters can be read by ComFlo:

xmin,	xmax,	ymin,	ymax,	zmin,	zmax
0.0	1.5	0.0	1.0	0.0	22.0
imax,	jmax,	kmax	dim		
32	1	128	2		
tmax	dt	fr-tot	tstart		
100.0	1.0E-2	10	0.0		
		freeslip.			
nu	domain	fdomain	fobject	f3D	
3.33E-3	11	0	0	0	
eps	omega	itmax	coef1	coef2	
1.0E-5	1.3	5000	1.5	-0.5	
object	radius	angle	var3	var4	
6	0.0	0	0.5	3.0	
posx	posy	posz			
0.0	0.0	22.0			
px	py	pz	bifstrt	avgstart	
20	20	100	4.0	150.0	
xp	yp	zp			
0.5	0.0	11.0			
alfax	alfay	alfaz			
1.0	1.0	1.0			
#ip	0(4)	exact	loadfld		
1	1	1	0		

Most of the names of the parameters can be found in the names of the common block variables, which are abstracted versions. The parameters `domain` and `object` define the chosen domain and object. In our case the backward facing step at $Re = 150$ with $H/S = 1.5$ is a tube as domain. The object is the backward facing step with $var3 = l$ and $var4 = S$. For more details about how the parameters should be chosen for a specified domain and object see the program ComFlo.

COMFLO-FLD

The file `comflo-fld` contains the complete velocity and pressure field at some time t and is used for back-up and/or as a start-up file.

COMFLO-ITER

The file `comflo-iter` shows the supremum norms of the velocity components u , v , w and the pressure of two successive time steps. Also it gives at the end the averaged number of iterations that each cycle required and the total number of iterations.

COMFLO.#.M and COORD.M

The file `coord.m` gives the coordinates in vector form that can be read by the program MATLAB. The files `comflo.#.m` give the velocity fields of u , v and w and the pressure field p , where "#" is a number. These fields are in matrix form and can also be read by MATLAB. In the program MATLAB we can visualize the data.

COORDX-, COORDY-, COORDZ-, COMFLO.#- and GEOMETRY.DAT

The files `coordx-`, `coordy-`, `coordz-` and `geometry.dat` are the AVS-equivalent of the MATLAB-file `coord.m`. The file `comflo.#.dat` is the AVS-equivalent of the MATLAB-file `comflo.#.m`.

Bibliography

- [1] B.F.Armaly, F.Durst, J.C.F.Pereira and B.Schönung, Experimental and theoretical investigation of backward-facing step flow, *Journal of Fluid Mechanics*, 127:473, 1983
- [2] R.B.Dean, Reynolds number dependence of skin friction and other bulk flow variables in two dimensional rectangular duct flow, *Journal of Fluids Engineering*, 100:215, 1978
- [3] J.Dijkstra, Simulation of Flow past Complex Geometries using Cartesian Grids, *Master's thesis*, RuG, 1997
- [4] J.Gerrits, Fluid Flow in 3D Complex Geometries - A Cartesian Approach, *Master's thesis*, RuG, 1996
- [5] P.Gresho, D.K.Gartling, J.R.Torczynski, K.A.Cliffe, K.H.Winters, T.J.Garret, A.Spence and J.W.Goodrich, Is the steady viscous incompressible flow over a backward-facing step at $Re=800$ stable? *Int. J. Num. Methods in Fluids*, 17:501, 1993
- [6] H.W.Hoogstraten, Stromingsleer, *Lecture notes*, RuG, 1992
- [7] L.Kaikitsis, G.E.Karniadakis and S.Orszag, Onset of three-dimensionality, equilibria, and early transition in flow over a backward-facing step, *Journal of Fluid Mechanics*, 231:501, 1991
- [8] J.Kim, P.Moin and R.Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, *Journal of Fluid Mechanics*, 177:133, 1987
- [9] K.Morgan, J.Periaux and F.Thomasset (Eds.), Analysis of laminar flow over a backward facing step: A GAMM Workshop, (*Notes on numerical fluid mechanics*; Vol 9), Vieweg 1984
- [10] A.E.P.Veldman, Numerieke Stromingsleer, *Lecture notes*, RuG, 1994
- [11] R.W.C.P.Verstappen and A.E.P.Veldman, Spectro-consistent discretization of Navier-Stokes: a challenge to RANS and LES, *J. Engineering Mathematics*, 34:163-179, 1998
- [12] P.T.Williams and A.J.Baker, Numerical simulations of laminar flow over a 3D backward-facing step, *Int. J. Numer. Methods in Fluids*, 24:1159, 1997