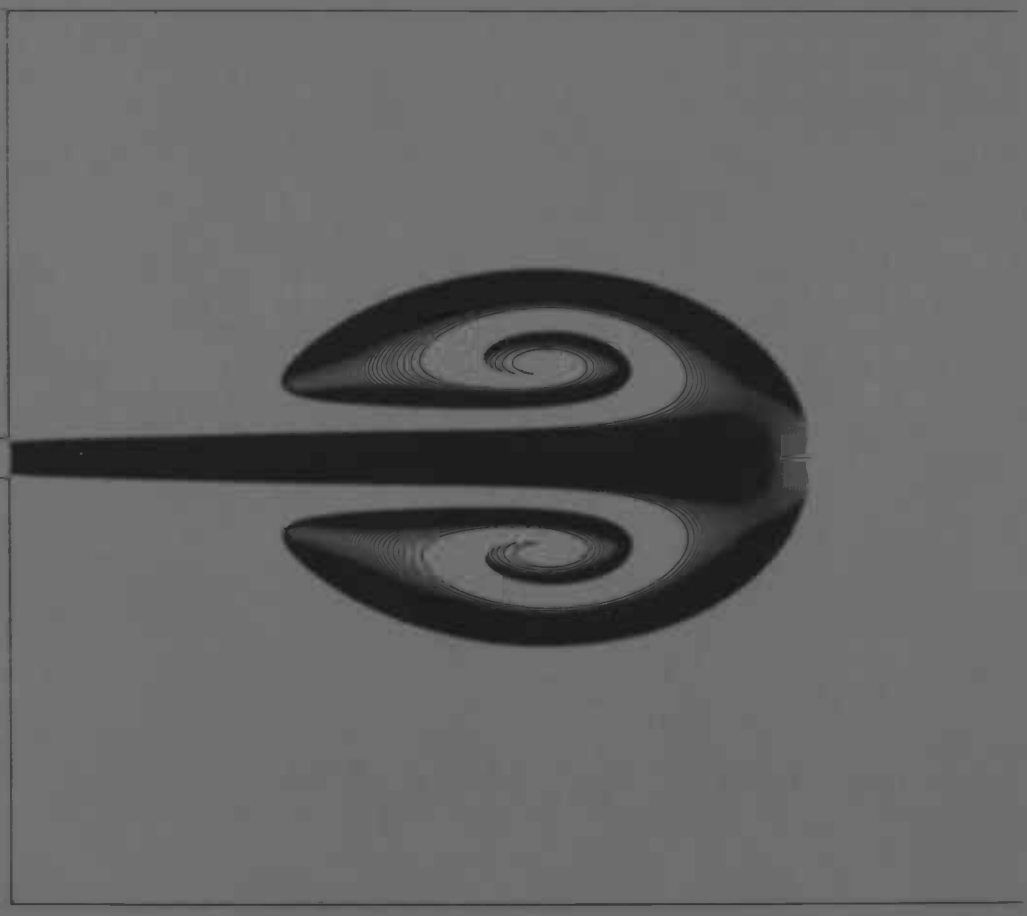
A simulation of vortexformation of a starting flow from a square-edged nozzle

A model for the human vocal cords

J.W. Phylipsen



Department of Mathematics

RuG

Master's thesis

A simulation of vortexformation of a starting flow from a square-edged nozzle

A model for the human vocal cords

J.W. Phylipsen

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

August 1996

Preface

This paper has been written as a master thesis under the direction of Dr. Ir. R.W.C.P. Verstappen at the University of Groningen.
The work was initiated by the University of Eindhoven, to validate the results of a computer program, which simulates a starting flow that leaves a square-edged nozzle. This computer program in turns is part of a Ph.D. thesis, that will be completed next year.
This work could not have been completed without the help of several people. First of all I would like to thank my supervisor Dr. Ir. R.W.C.P. Verstappen for all his explanations and support, furthermore Prof. Dr. A.E.P. Veldman for his fruitful discussions, B. de Groot for all his help and efforts to make me understand Matlab and P.J. van Daal for making this thesis into a readable piece of work. I would also like to thank all the people, I didn't mention, for al their direct and indirect support.

Groningen, 29 augustus 1996
Jan-Willem Phylipsen.

Summary

In 1993 at the University of Eindhoven, as part of a Ph.D. thesis, research was done on how sound is produced in a flue organ. On this very moment, at the University of Eindhoven as well, research is done on the behavior of less viscous flows that leave a square-edged nozzle. Using a computer program, a simulation of the behavior of less viscous flows was done. This research is done to get a clearer view on how the human vocal cords work. People, who as a result of larynxcancer, do not have their vocal cords anymore, can only speak by burping up air and using 'electronic' vocal cords. These people are much helped by this research.
The aim of this master thesis is to validate the numerical results of the computer program, developed in Eindhoven. However, because the Ph.D. thesis takes up more time than this master thesis, we will only be able to validate the results of our program with experimental results.

Contents

Chapter 1

Introduction

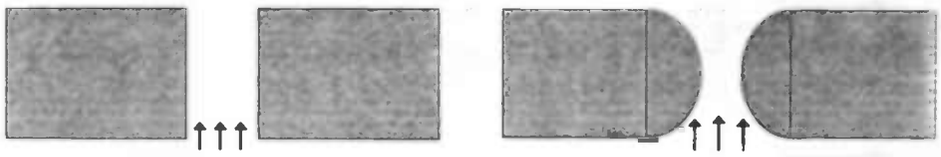
1.1 Vortex-blob Method

In 1993 at the University of technology Eindhoven a Ph.D. thesis was completed on 'Aeroacoustic sources in internal flows'. One of the subjects in this thesis was, how sound is produced in a flue organ. An answer to this question was found in using a numerical method, implemented in a computer program, called the vortex-blob method. This method only models the convection in the flow, and doesn't deal with diffusion. For non viscous fluids this method has proven to give astonishing good results. However, when we want to model viscous flows like water or even air the vortex-blob method fails, because we will have to pay more attention to the diffusive part. Therefore a diffusive vortex-blob method has been developed at the University of Eindhoven, one that gives good results for viscous flows. The enhanced vortex-blob method originated from the Navier-Stokes equations and exactly these equations play the leading part in this master thesis as well.
The department of Applied Physics in Eindhoven has filed an application to the department of Computational Mechanics at the Rijksuniversiteit Groningen to validate the results obtained by a computer program, using the diffusive vortex-blob method. This program simulates how vortices develop in time, as a result of a starting flow from a square-edged nozzle. The research on this subject is again part of a Ph.D. thesis.

1.2 Aim of this Thesis

This simulation of vortexformation is a model for the human vocal cords. The human vocal cords are in fact two muscles, through which air flows, that produce our speech. By stretching our vocal cords we can produce different sounds. When the ability to let our vocal cords vibrate is lost, for example because of larynx cancer, one must use an electronic vibrator. The speech produced with such a

vibrator sounds very robot-like and emotionless. In order to improve the speech of these people and give them a more human voice, it is important to know how a flow expands when it leaves a square-edged nozzle. Mainly the transitional stage from laminar to turbulent flow is of much interest. From our speech we know that, if the human voice breaks, the soundfrequency graph show a discontinuity. When we look at the flow we see that it is exhibiting turbulent behavior. Two geometrical models of vocal cords are shown in Figure 1.1.



(a) geometry 1 (b) geometry 2

Figure 1.1: Two geometrical models of the vocal cords

The calculations are done using the geometry in Figure 1.1(a), the arrows in Figure 1.1 indicate the direction of the flow. Using this geometry, we shall focus on two cases. This research shall concentrate on

1. **Mathematical model.** What are the equations we must solve?
2. **Numerical solution.** How are these equations implemented in a computer program? What testcases were carried out to validate the results of the DNS program?
3. **Numerical results for case I.** Some characteristic plots of velocity, pressure, vorticity and streaklines for a typical case.
4. **Numerical results for case II.** Same as above but with a time-dependent inflow.
5. **Conclusions & recommendations.** What can we say about our model and what is there more to research?

In short, the exact aim of this thesis is to validate the results of the diffusive vortex-blob method, implemented in a computer program. This validation is done by a computer program as well, which solves the same equations, but with a different method. As we mentioned before, the results of the computer program developed in Eindhoven must be validated with the results our computer program. However, because the Ph.D. thesis takes up more time than this master thesis, we will only be able to compare the results of our program with experimental results.

Chapter 2

The Mathematical Model

2.1 Introduction

In order to be able to create a workable model, some assumptions were made. One was concerning the grid. We expected that for an accurate solution the number of gridlines in the horizontal direction, nx, had to outnumber the number of gridlines in the vertical direction, ny. The hart of the DNS[1] program uses the ICCG[2]-method as Poisson solver, which in turns uses some assumptions concerning its input. The ICCG-solver will only work when $nx \leq ny$, a choice that was made when the DNS program was developed a few years ago. In order to let the ICCG-solver work with our new restriction, we simply flipped the x,y-axis, the x-axis now becoming vertical and the y-axis horizontal. The ICCG assumption is then satisfied, but, for example, the traditional horizontal velocity-component u is now pointing in vertical direction. In this master thesis we shall refer to v as the velocity-component pointing in the horizontal direction of the outgoing flow and to u as the orthogonal component to this velocity.

2.2 The Geometry

The University of Eindhoven has requested us to use the geometry in Figure 2.1 in our model. We can see that the channel is symmetrical round the line x=$\frac{1}{2}$. Using this symmetry the computation time could be reduced with a factor 2. Hovever, we did not use this option, because we wanted to keep the DNS program as general as possible. Now we could change the geometry as we wished, without having to change the hart of the program.

[1] Direct Numerical Simulation
[2] Incomplete Choleski Conjugent Gradient

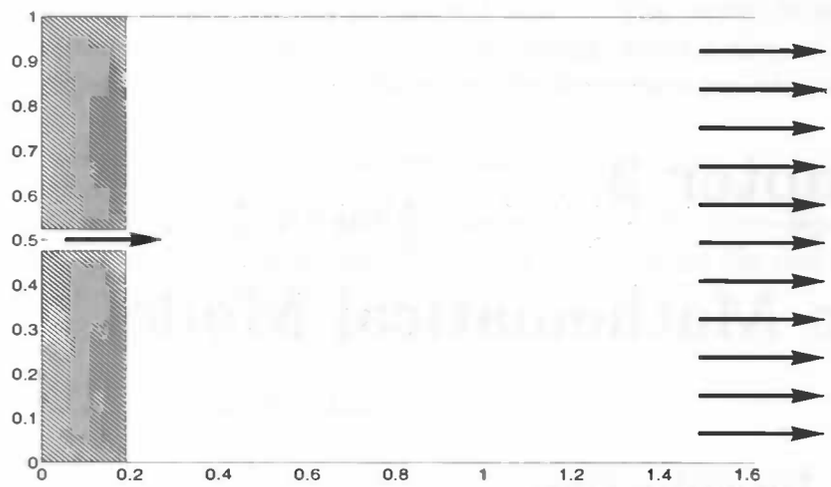


Figure 2.1: The geometry used for the computation of case I

2.3 Navier-Stokes Equations

We will consider an incompressible flow, that is we assume the density of the flow to be constant in space and time. In our model the viscosity of flows is constant, flows with this property are usually called *Newton flows*. The entire model can be described with the *Navier-Stokes equations*. These equations are in fact a combination of two equations:

- the *continuity equation*, which describes the mass-balance of the flow.
- the *momentum equations*, which determines the impuls-balance.

In the next paragraphs each equation will be discussed.

2.3.1 Continuity Equation

The continuity equation describes the mass-balance. Because the fluid is incompressible, the density ρ has vanished out of the equation and we get

$$\nabla \cdot \vec{v} = 0 \tag{2.1}$$

Note that $\vec{v}$ is the vector $\begin{pmatrix} u \\ v \end{pmatrix}$

2.3.2 Momentum Equations

The non-dimensional momentum equations are

$$\frac{\mathrm{D}\vec{v}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta\vec{v} \tag{2.2}$$

The capital D in the equation refers to the material derivative, defined as

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \vec{v}\cdot\nabla \tag{2.3}$$

However, we will use the non-dimensional equations in which we have introduced a new parameter *Re*, the Reynolds number. The coordinates x and y are scaled with the heigth H of the small input channel shown on the left in Figure 2.1 and the velocities u and v are scaled with the Bernoulli velocity U_B defined as $\sqrt{\frac{2\Delta p}{\rho_{CO_2}}}$. This velocity is deduced from the Bernoulli equation $p + \frac{1}{2}\rho u^2 = \text{constant}$. *Re* is generally defined as $\frac{Ul}{\nu}$, where U is a characteristic velocity, l a characteristic length and ν the kinetic viscosity of the medium. Substituting H and U_B in the formula for *Re*, we get $\frac{U_B H}{\nu}$. The equations can now be written as

$$\frac{\mathrm{D}\vec{v}}{\mathrm{D}t} = -\nabla p + \frac{1}{Re}\Delta\vec{v} \tag{2.4}$$

These equations contain the following quantities:

- two velocity components, u and v.
- the pressure, p.
- the Reynolds number, *Re*.

2.4 The Model

Written out in components equation 2.3 reads:

$$\begin{cases} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{cases} \tag{2.5}$$

In the first and the second equation we can distinguish four terms. The first term with the time derivative is the instationary part, the second term is the convective part, the third term on the righthand side is the pressure gradient and the fourth part is the diffusive component of the equation. The input-velocity has a discontinuous profile, u=v=0 on the wall and u=0, v=1 inside the channel. To make the model unequivocal, two boundary-conditions are neccesary. Because a

viscous medium is modelled the *no-slip* condition is the first boundary-condition. This condition states that the velocity on the wall is zero. The second boundary-condition is a Neuman condition on the outflow boundary. Several choices can be made, but we have chosen for $\frac{\partial^2 v}{\partial y^2} = 0$. Therefore the boundary-conditions are:

- the no-slip condition: $u = v = 0$ on each wall.
- the outflow boundary: $\frac{\partial^2 v}{\partial y^2} = 0$ and $\frac{\partial p}{\partial y} = const$, where the time-dependent constant is determined such that the in-flux at $y = 0$ equals the out-flux at the out-flow boundary at time t.

2.5 The Parameter: *Re*

The parameter *Re* determines the relation between velocity, viscosity and length. For example, when a new wing profile is simulated by computer, Reynolds numbers in order of 10^7 are used. At the moment we can solve a problem in a reasonable amount of time with Reynolds number 10^5 using a DNS program. For this problem however, smaller Reynolds numbers are used. Two cases particulary are observed, using $Re = 1360$ and $Re = 5127$.

Chapter 3

The Numerical Solution

3.1 The Algorithm

The Navier-Stokes equations are convection-diffusion equations. We shall call the convective part a(**u**), and the diffussive part b(**u**). Using these abbreviations the Navier-Stokes equations can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + a(\mathbf{u}) = -\nabla p + \frac{1}{Re} b(\mathbf{u}) \tag{3.1}$$

To explain how this algorithm works, a semi-discretization in time is used. The convective part is treated with a modified Adams-Bashford method. The details of the Adams-Bashford method can be found in [2]. This now results in the following semi-discretization of the Navier-Stokes equations

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\delta t} + a(\frac{3}{2}\mathbf{u}^{n} - \frac{1}{2}\mathbf{u}^{n-1}) = -\nabla p^{n+1} + \frac{1}{Re} b(\mathbf{u}^{n}) \tag{3.2}$$

n is the time-index and δt is the timestep. The semi-discretization of the continuity equation is straightforward

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{3.3}$$

The time-index must be n+1, because a velocity calculated at time-level n would never satisfy the continuity equation with an index n. The algorithm now consists of three parts

1. integration of the velocities without the pressure component.
2. calculation of the corresponding pressure by satisfying the continuity equation.
3. adjusting the velocities by using the calculated pressure.

We shall now discuss each step of the algorithm.

3.1.1 The Algorithm: Step 1

Consider the following discretization

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\delta t} = -a(\frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1}) + \frac{1}{Re}b(\mathbf{u}^n) \tag{3.4}$$

From equation (3.4) $\tilde{\mathbf{u}}^{n+1}$ can be computed since all other velocities at time-level n and n-1 are known.

3.1.2 The Algorithm: Step 2

This part of the algorithm demands the majority of the computation time. Consider the following equation

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\delta t} = -\nabla p^{n+1} \tag{3.5}$$

Equation (3.5) contains two unknowns: $\mathbf{u}^{n+1}$ and p^{n+1}. Since the continuity equation is valid in the entire medium, we will apply (3.3) to (3.5). This will result in the following equation

$$-\frac{\nabla \tilde{\mathbf{u}}^{n+1}}{\delta t} = -\Delta p^{n+1} \tag{3.6}$$

From (3.6), p^{n+1} is calculated using ICCG. This is a Poisson-solver implemented a few years ago at the department of Mathematics of the University of Groningen and is found to be very succesfull.

3.1.3 The Algorithm: Step 3

The last step calculates the velocities at the new time-level using equation (3.5). Now we have solved one timestep of the Navier-Stokes equations. It may be clear that by substituting equation (3.5) in equation (3.4), $\tilde{\mathbf{u}}^{n+1}$ can be eliminated and equation (3.2) is solved. This method must use the temporary variable $\tilde{\mathbf{u}}^{n+1}$, because p is implicitly given in equation (3.2). We always try to use as much implicit terms as possible, because these terms are unconditionally stable, but they are not very suitable for parallel computers. Stability will be discussed in section 3.4.

3.2 Adjustments to the geometry

As is stated in Section 2.2, we wanted to keep the DNS program as general as possible. The original DNS program carried out its calculations on a rectangle with Dirichlet boundary conditions. We adjusted the DNS program at two points to make it suitable for this problem:

- Two pieces were cut out of the computational domain, see Figure 2.1.
- One Dirichlet boundary condition was changed into a Neuman boundary condition, namely the outflow boundary. This outflow boundary condition can be found on page 12.

On the two pieces, that were cut out of the computational domain, the equations of model 2.5 are still computed, but in such a way that do not interfere with the calculations in other points: the pressure is set to unity in the virtual part of the computational domain and the velocity is set to zero.

3.3 The Grid

In order to discretize (2.5) in space, we have used FVM[1]. This method uses a grid for its calculations. Therefore the geometry in Figure 2.1 is covered with a stretched grid of 100 by 100 gridlines, displayed in Figure 3.1.

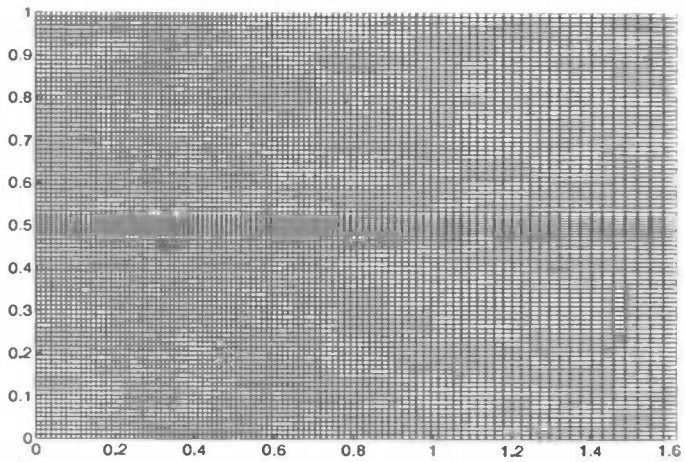


Figure 3.1: 100 by 100 stretched grid, used for the computation

Since we expect large velocity variations, the number of gridlines is increased near the wall, because the fluid might become turbulent in this area. Just as the flow leaves the square-edged nozzle, great vorticity is to be expected, therefore we have taken more gridlines near the nozzle and less gridlines the more we move to the end of the channel. The grid is stretched using a tangent function, with a parameter for adjusting the extent of the stretching. The stretching is defined as $\frac{\max(dx)}{\min(dx)}$. In Figure 3.1 the stretching in the x-direction is 1.8 and in the y-direction 3.0.

[1] Finite Volume Method

3.4 Discretization & Stability

3.4.1 Discretization

As stated in the previous section, FVM has been used to solve (2.5). In short, we will discuss this method, but more details can be found in [9]. Several choices can be made concerning the position of the variables in the grid. We have chosen the Marker-And-Cell (MAC) method. In this method, the velocities u and v and the pressure p are uncommonly defined than we are used to, Figure 3.2 shows how.

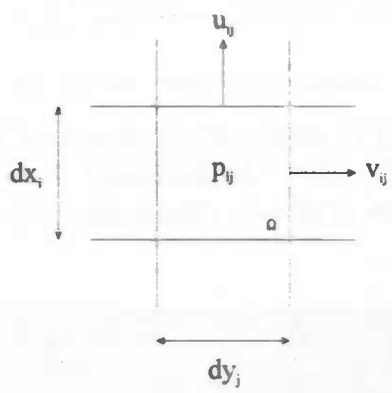


Figure 3.2: Staggered locations of u, v and p, as in MAC

The x-momentum equation is applied to the point where u is defined and for the y-momentum equation to the point where v is defined. All pressure points are connected with eachother excluding any unwanted oscillations. More details on this subject can be found in [9]. The discretization of (2.5) is done using the conservative form

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(u^2) + \frac{\partial}{\partial y}(uv) = -\frac{\partial p}{\partial x} + \frac{1}{Re}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \tag{3.7}$$

Equation (3.7) is only valid in the x-direction, an equation in the y-direction is analogue. First we use step one of the algorithm: integration of (3.7) over Ω without the pressure component

$$\int_\Omega \frac{\partial u}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \nabla \cdot \begin{pmatrix} u^2 \\ uv \end{pmatrix} \mathrm{d}\Omega = \frac{1}{Re}\int_\Omega \nabla \cdot \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} \mathrm{d}\Omega \tag{3.8}$$

We now apply the divergence theorem of Gauss to (3.8). This theorem will transform a surface-integral into a contour-integral. In this contour-integral a variable **n** arrises, which is the normalvector at the side of a cell over which is integrated. This vector always points outwards. More on Gauss can be found in [1]. In (3.9) S is the edge of a cell, also called $\partial\Omega$

$$\frac{\partial}{\partial t}\int_{\Omega} u \,\mathrm{d}\Omega + \int_{\partial\Omega} \begin{pmatrix} u^2 \\ uv \end{pmatrix} \cdot \mathbf{n}\,\mathrm{d}S = \frac{1}{Re}\int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}}\,\mathrm{d}S \tag{3.9}$$

Note that Fubini's theorem, as given in [7], may be applied to (3.9) since we assume only C^{∞}-functions to be a solution of (2.5). Now the only equation to discretisize is the continuity equation (3.3). Again we integrate over Ω and apply Gauss

$$\int_{\Omega} \nabla \cdot \begin{pmatrix} u \\ v \end{pmatrix}^{n+1} \mathrm{d}\Omega = 0 \Leftrightarrow \int_{\partial\Omega} \begin{pmatrix} u \\ v \end{pmatrix}^{n+1} \cdot \mathbf{n}\,\mathrm{d}S = 0 \tag{3.10}$$

Now each integral can numerically be approximated for example by using midpoint, giving a second order result. A full deduction of all discretizations goes beyond the context of this thesis and therefore we shall only discuss the discretization of the continuity equation. All the other equations are treated analoguous. First one chooses in which direction the edge must be followed, for example counter-clockwise. Starting at the right edge corresponds with $\mathbf{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, by taking the dot-product with the velocity vector, only $v_{i,j}^{n+1}$ remains. The right side of the cell, using midpoint, will be $v_{i,j}^{n+1}\mathrm{dx}_i$. Doing this for all the other sides of the cell will result in the following discretization of (3.10)

$$u_{i,j}^{n+1}\mathrm{dy}_j + v_{i,j}^{n+1}\mathrm{dx}_i - u_{i-1,j}^{n+1}\mathrm{dy}_j - v_{i,j-1}^{n+1}\mathrm{dx}_i = 0 \tag{3.11}$$

3.4.2 Stability

Consider the following set of equations, written in matrix form

$$\mathrm{A}x = b \tag{3.12}$$

Let **A** be a n×n-matrix and x and b vectors of length n. From [9] we know that (3.12) can be solved quickly if **A** is a K-matrix. The definition of a K-matrix is

Definition 1 *Matrix $(a_{i,j})$ is a K-matrix if*

1. $a_{i,i} > 0$
2. $a_{i,j} \leq 0 \;\forall\; i \neq j$
3. $a_{i,i} \geq -\sum_{i \neq j} a_{i,j}$ *and for at least one i a strict inequality*

We will now focus on the stability of equation (3.2) and equation (3.3). Equation (3.3) is unconditionally stable, since it is implicit. We are now looking for a limitation of the time-step in order to keep the method converging. We have

carried out a stability analysis for (3.2) without Adams-Bashford but for simple Euler, i.e. for

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} + a(\mathbf{u}^n) = -\nabla p^{n+1} + \frac{1}{Re} b(\mathbf{u}^n) \tag{3.13}$$

Stability analysis on Adams-Bashford is extremely complicated, and in practice the results of the stability analysis for Euler is slightly worse than an analysis on Adams-Bashford. As we can see in (3.13) the pressure gradient is implicit and therefore unconditionally stable. We will leave this component out of the analysis. The stability analysis is carried out in two steps and only for the velocity component u.

Step 1. stability with respect to the convective terms

Step 2. stability with respect to the diffusive terms

We shall now discuss each step seperately.

Convective Stability

Consider the following equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} \tag{3.14}$$

Because of the complexity of the stability analysis on non-lineair equations, we assume u and v, the coefficents of the partial derivatives, to be constant. Therefore we must look at equation (3.14) as a lineairization of (3.1). This equation is now discretisized with forward Euler in time and upwind in space. When we isolate $u_{i,j}^{n+1}$ on the left side, reorder all variables and use for $\delta x = \min_i \mathrm{dx}_i$ and $\delta y = \min_j \mathrm{dy}_j$, we get

$$u_{i,j}^{n+1} = (1 - \frac{u\delta t}{\delta x} - \frac{v\delta t}{\delta y})u_{i,j}^n + \frac{u\delta t}{\delta x}u_{i-1,j}^n + \frac{v\delta t}{\delta y}u_{i,j-1}^n \tag{3.15}$$

When we apply definition 1 to (3.15), we find the following limitation for the time-step

$$\delta t \leq \frac{\delta x \delta y}{u\delta y + v\delta x} \tag{3.16}$$

Diffusive Stability

Consider the following equation

$$\frac{\partial u}{\partial t} = \frac{1}{Re}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \tag{3.17}$$

This equation is treated with forward Euler in time and central in space. When we reorder as before, we get

$$u_{i,j}^{n+1} = \frac{\delta t}{Re\delta x^2}(u_{i+1,j}^n + u_{i-1,j}^n) + (1 - \frac{2\delta t}{Re\delta x^2} - \frac{2\delta t}{Re\delta y^2})u_{i,j}^n + \frac{\delta t}{Re\delta y^2}(u_{i,j+1}^n + u_{i,j-1}^n) \tag{3.18}$$

Again we apply definition (1) to (3.18) and we will get the following limitation of the time-step with respect to the diffusive terms

$$\delta t \leq \frac{\delta x^2 \delta y^2 Re}{2\delta x^2 + 2\delta y^2} \tag{3.19}$$

We may now conclude that for the stability of the numerical method we must use the time-step that is the smallest of (3.16) and (3.19). In the aircraft industry and related industries, larger Reynolds numbers are used. As we look at (3.16) and (3.19) we can see that the convective part of the Navier-Stokes is dominant above the diffusive part. In practice it is the convective component that determines the limitation of the time-step.

3.5 Test Cases

When all modifications were made to the DNS program, some tests were done to ensure the correctness of the program's results. We have chosen the following tests, since we knew what the output should be.

1. When the input-velocity of a channel of length 3 is 1, its output-velocity should be a Poiseuille flow.

2. If we take Figure 2.1 and let the velocity in the entire geometry be 1 and we give all walls a velocity of 1 as well, then the velocity of the flow must remain 1.

We will start with test 1. The input-velocity has a discontinuous profile, u=v=0 on the wall and u=0, v=1 inside the channel. This profile will gradually change into a Poiseuille profile, however, the rate of change is dependent on the Reynolds number. The Reynolds number used for this test was 100, which is considerably smaller than the cases mentioned in Section 2.5. For this test we used a 12×40 grid. We can see in Figure 3.3 that a Poiseuile flow arises after a few time-steps. In test 2 the initial velocity is u=0 and v=1 everywhere in the geometry, even on the walls. This grid has dimensions 20×40, and five gridlines in the small channel. After 150 time-steps the vector velocity field has not changed; see Figure 3.4. Therefore we may conclude that the calculation of the velocities is correct.

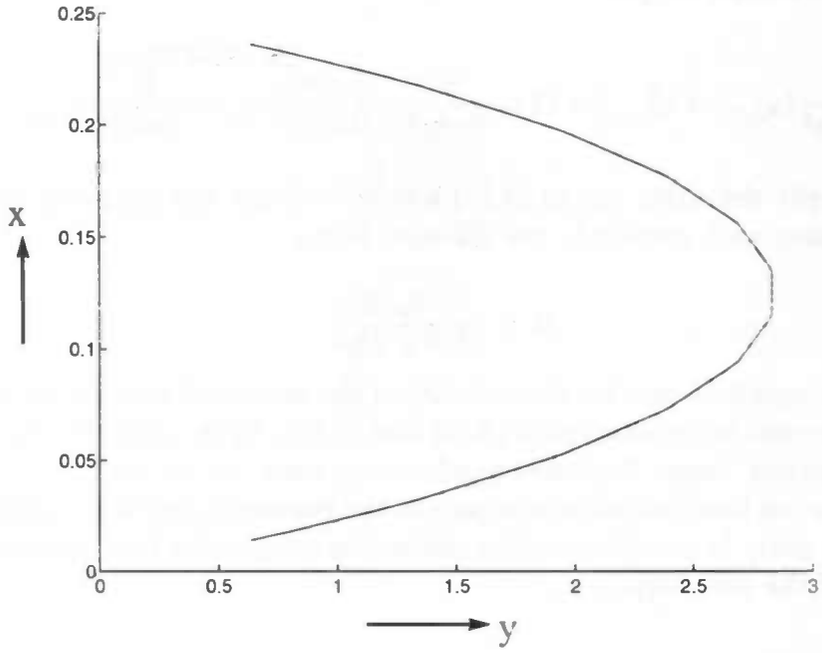


Figure 3.3: Contourplot of testcase 1, a Poiseuille flow

3.5.1 Test case for the streaklines

In the previous section, test cases were carried out for the calculation of the velocities, but we have added an extra subroutine to the DNS-program computing the streaklines of the flow. This subroutine also contains an extrapolation routine for the following reason: a particle that is followed during the simulation might start in a point where its velocity is exactly known. One time-step later however, the particle need not still be in a gridpoint, but it could have moved randomly inside a cell. At this exact moment, we must apply the extrapolation formula, that changes for each quadrant of the cell. At the end of the subroutine we integrate the velocities, using Euler's method. More details on this routine can be found in the subroutine 'cntour' of the DNS-program. To check whether this subroutine gives correct results, the following test has been done.
Three particles were followed in such a way that they would cross each quadrant of a cell. This was done to test each extrapolation formula, and the transition from one quadrant to the other. The initial velocity is $u = \frac{1}{4}$, $v = \frac{3}{4}$ and it must not change, since only the subroutine, in which the streaklines are computed, is called. Figure 3.5 shows how these particles move in time. They move along a straight line, indicating a correct implementation.

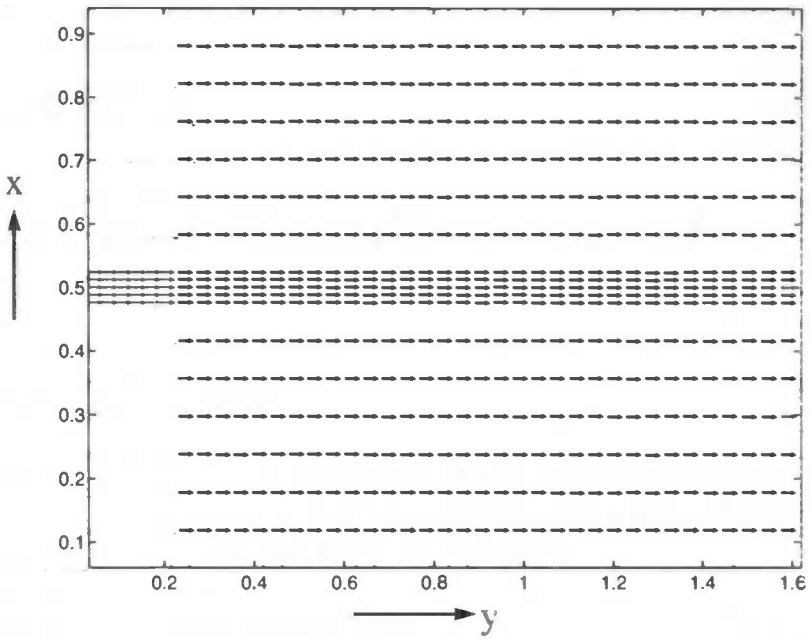


Figure 3.4: Vector velocity plot of test case 2

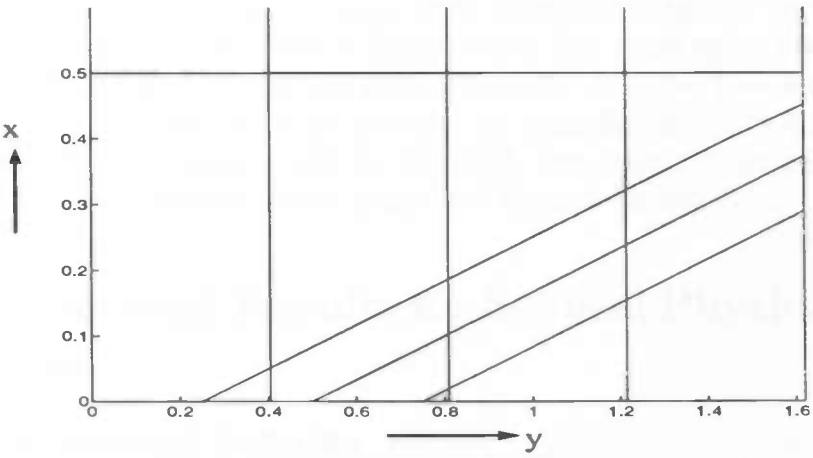


Figure 3.5: Grid 2×4, test for the extrapolationformula

Chapter 4

Numerical Results - *case I*

4.1 Introduction

In this chapter we will look at the results of the numerical solution of (2.5). For simple flows we can calculate an analytic solution, but a general analytic solution has not yet been found. We will focus on two cases

- case I: Re=1360 with steady inflow
- case II: Re=5127 with unsteady inflow

The geometry used for case I was shown in Figure 2.1. For the second case we used a slightly different geometry. Apart from the Reynolds number the difference between these two cases is that case I has an inflow with constant velocity and case II has a time-dependent inflow. The second case will be discussed in the next chapter. Some experiments have been carried out by the department of Applied Fysics at the University of Eindhoven a few years ago. The results of these experiments can be found on page 105 of [5]. For case I we studied three time-steps, corresponding with the pictures on page 105 of [5]. In the following sections four types of graphs will be depicted, the velocity, the pressure, the vorticity and the streaklines. Each graph will first be discussed.

4.2 Numerical Results for Several Physical Quantities

4.2.1 Numerical Velocity

Since the velocity is solved from the Navier-Stokes equations on a strectched grid is used the components, u and v are not defined in the same gridpoint. By interpolation we obtain a velocity grid defined in each gridpoint. In each

gridpoint a horizontal and a vertical velocity can be combined resulting in a velocity vector. For *Re*=1360 three vector-velocity plots have been made at three different time-steps.

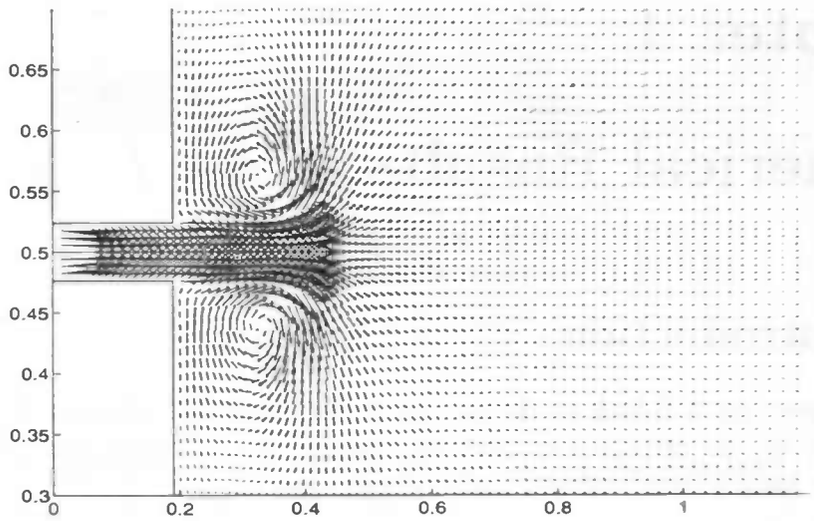


Figure 4.1: Vector-velocity plot at t=0.48 and *Re*=1360

In Figure 4.1, 4.2 and 4.3 we can clearly see the two vortices moving to the right of the geometry. When we look more closely we can see that a second flow is initiated. As a result of the viscosity of the fluid, the fluid above and below the square-edged nozzle is moving towards the main flow. The maximum velocity occurs on the centerline $x = \frac{1}{2}$, and the velocity in the center of the vortices is zero.

4.2.2 Numerical Pressure

The pressure within the flow is calculated at each time-level and is renormalised by substracting its mean value if the mean pressure is greater than 10^{-10}. With respect to the pressure we can look at three interesting plots

- the pressure drop along the centerline $x = \frac{1}{2}$
- the pressure on the wall $y = \frac{4}{21}$
- the pressure distribution on the lower wall $x = 0$

Since we have a symmetrical sollution, a plot of the pressure on the upper wall $x = 1$ will be the same as on the lower wall. The plots were made using the same

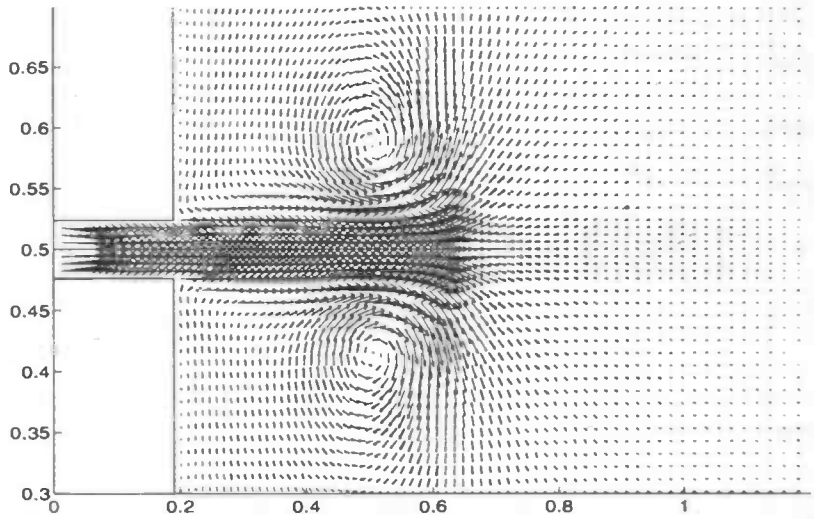


Figure 4.2: Vector-velocity plot at t=1.00 and Re=1360

time-steps as for the velocity plots at Re=1360. First we will pay attention to the first series of plots. The results are shown in Figure 4.4.
In each graph we can see a slump at $y = \frac{4}{21}$, indicating the position of the square-edged nozzle. Further along the x-axis we first see a bigger slump followed by a hump. The second slump roughly indicates the position of the center of the vortices - this is a rough indication since we observe the pressure along the centerline and not along a straight line through the center of the vortex. The hump reveals the position of the front. In that point, two particles coincide, namely particles from the main flow and particles rotating in the vortices. This is where they come together and induce a local pressure-bump. As we move on to the right we see that the pressure tends to zero.
Now we will discuss the next series of plots concerning the pressure distribution along the line $y = \frac{4}{21}$. Figure 4.5 shows two points in which the pressure is minimal. These points correspond to the position of the square-edged nozzle. The last series of plots show the pressure on the lower wall.
In Figure 4.6 we can also see the position of the square-edged nozzle clearly. At $x = \frac{4}{21}$ we see a sudden drop in pressure, followed by a gradually increasing pressure as we move on to the right. At the end of the geometry we can see the pressure suddenly dropping. This however, is not a physical phenomenon. To ensure mass-balance, we have subtracted a value from the value of the velocity on the right side of the geometry, so $Q_{in} = Q_{out}$ would be satisfied. Q is the mass-flux defined as the integral of velocity over the length of the local cross-section of the geometry. Therefore a sudden change in velocity will have a direct effect on the pressure. Should we not apply this modification, the poisson-solver would

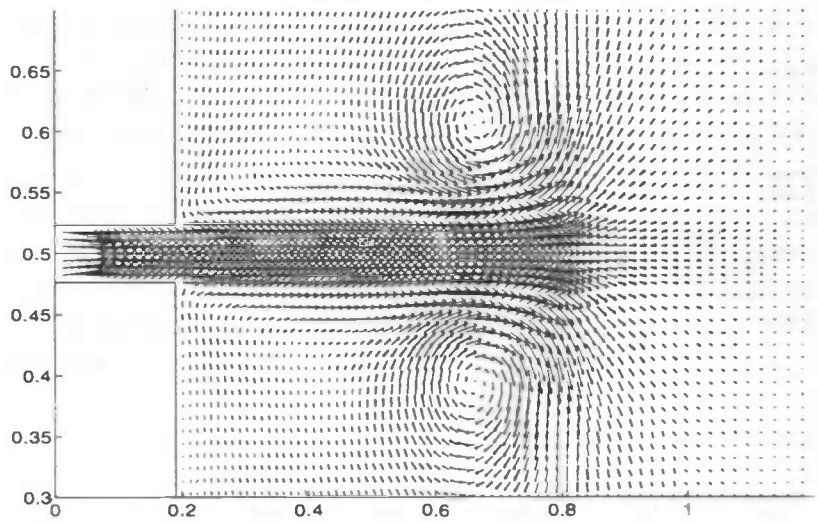


Figure 4.3: Vector-velocity plot at t=1.52 and Re=1360

not work properly and could even diverge. Note that in Figure 4.4 and Figure 4.6 the pressure gradient over the heigth of the outflow boundary is constant.

4.2.3 Numerical Vorticity

The vorticity or rate of rotation is defined as

$$\Omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{4.1}$$

If the rate of rotation in the flow equals zero everywhere, the flow is called rotation-free. When we look at Figure 4.2 it is obvious that our flow is not rotation-free. By discretization of (4.1) we obtain a numerical approximation of the vorticity in the geometry. Figure 4.7 shows the evolution of the vorticity for Re=1360.

The vorticity is zero at the center of the circles and increases or decreases the more we move outwardbound. If the local rotation in the flow is counter-clockwise, then the vorticity is negative; clockwise rotation gives a positive vorticity.

4.2.4 Numerical Streaklines

All the previous plots were indirect indications of activities in the flow. Now we will discuss the streaklines of the flow, which gives in fact the actual flow visualization. The streaklines for Re=1360 are shown in Figure 4.8.

How where these pictures created? We have followed 100 particles during 380 time-steps. At each new time-step a new serie of 100 particles is followed. This means that in Figure 4.8(c) $100 \times 380 = 38.000$ points are drawn. The points on the right of the narrow channel were followed 1 time-step and the points at the front of the graph were followed 380 time-steps. This is the part of the program where we use the extrapolation formula mentioned in the previous chapter. It may be obvious that these calculations take up a reasonable amount of calculation time. In this case, for each series of points, 100 calculations must be carried out. At time-step one, 100 calculations, at time-step two, 200 calculations etcetera. The amount of calculations is of second order with respect to the number of time-steps.
When we examine the plots in Figure 4.8 more thoroughly we see that the streaklines of the flow inside the narrow channel are condensed. When the flow leaves the square-edged nozzle the flow has more space and it will expand. Further downstream however, the streaklines are condensed again, because of the rotation of the flow. Because we simulate an incompressible flow, the density is constant everywhere, so when we compare the vector-velocity plots with the streaklines we see that the streaklines lie nearer to each other there where the vector-velocity plots indicate higher velocities.

4.2.5 Experimental Streaklines

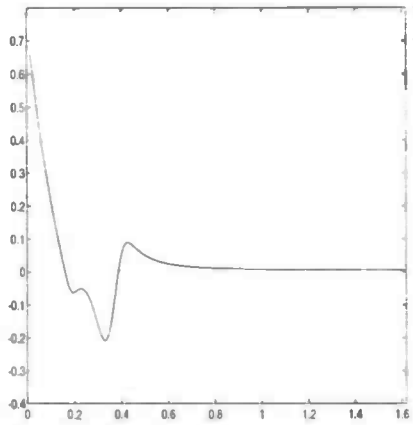
The pictures shown in figure 4.9 are copied from [5]. Fortunately we see great similarity with our pictures in Figure 4.8.
In Figure 4.9 we see how CO_2-gas expands, when it leaves a square-edged nozzle as a result of a constant pressure difference over the entire geometry. The very reason why we can see the expansion of the CO_2-gas in this figure, is that we see it's density distribution. However, one fundamental assumption in this thesis is that we observe an incompressible flow and therefore presume the density to be constant. This is one reason for some small differences between Figures 4.8 and 4.9. Another reason for differences is that for the pictures in Figure 4.9 a time-dependent inflow profile is used instead of a constant inflow profile. In Figure 4.9 we also see that the flow tends to move to the right, which is probably caused by a not entirely symmetrical geometry.

4.3 Calculation Time

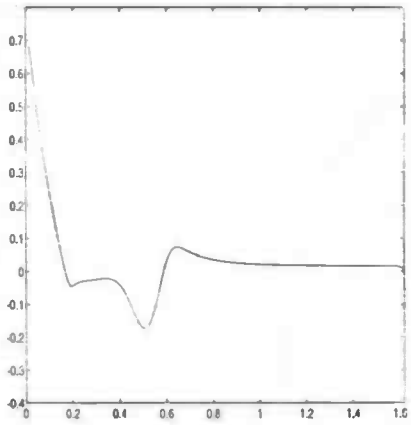
Some calculations were carried out on a Pentium-90 with 16 MB memory using a Fortran compiler of the University of Salford. This computer has an approximated peak performance of 13 Mflops[1]. The original DNS-program could only calculate the velocity and pressure. Using a 100×100 grid with 1000 time-steps the

[1] Million floating point operations per second

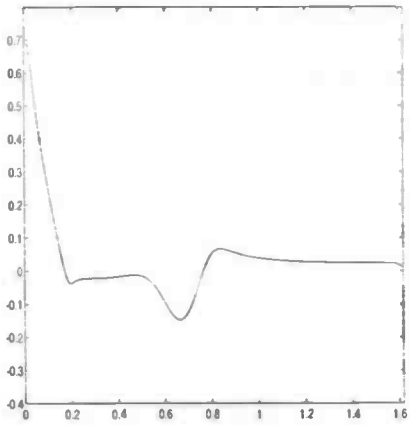
computation took about 20 minutes. After we adjusted the DNS-program for the calculation of streaklines, the computation time was roughly doubled. The computations were done at the University of Groningen as well, this time on a Hewlett Packerd 755. The HP has a peak performance of 40 Mflops. This computer however is part of a local network, which can affect the computation time. When more than three users are logged, the Pentium-90 beats the HP 755. One advantage of the HP 755 however, is that its Fortran compiler can be optimized, which can speed up things with a factor three. So, in conclusion the DNS-program is very suitable to run on PC's, since the DNS-program requires little memory. With 4 MB internal memory a 100×100 grid can be solved without any stack problems.



(a) t=0.48

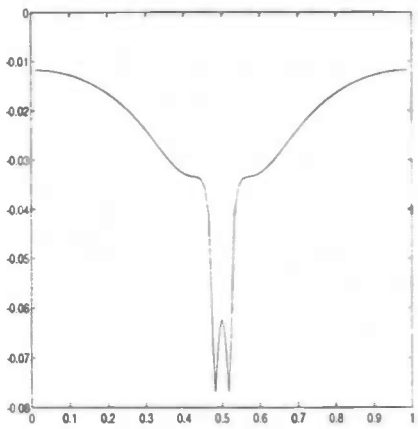


(b) t=1.00

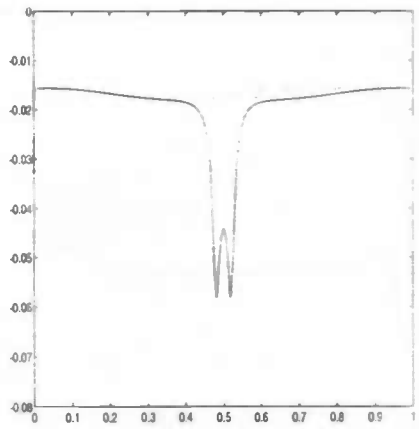


(c) t=1.52

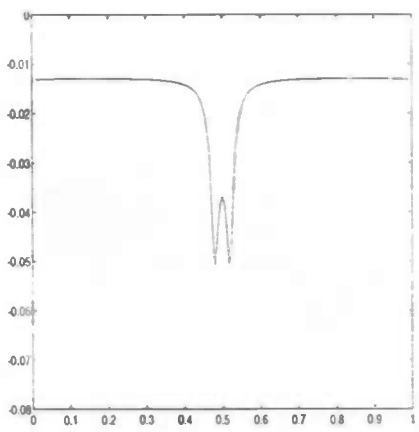
Figure 4.4: Pressure-plot along the line $x = \frac{1}{2}$ for Re=1360



(a) t=0.48

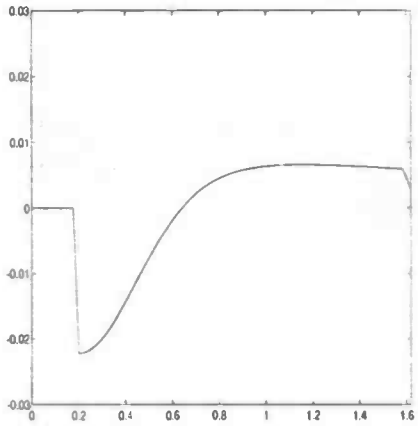


(b) t=1.00

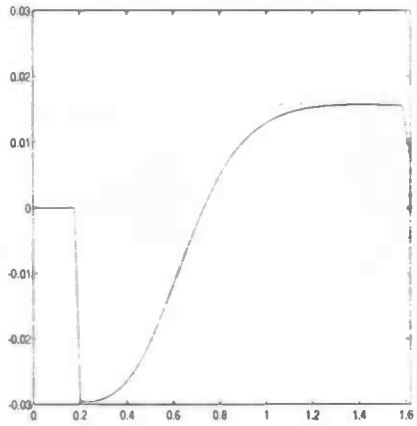


(c) t=1.52

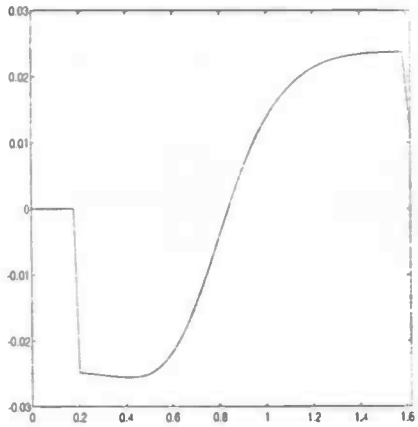
Figure 4.5: Pressure-plot along the line $y = \frac{4}{21}$ for Re=1360



(a) t=0.48

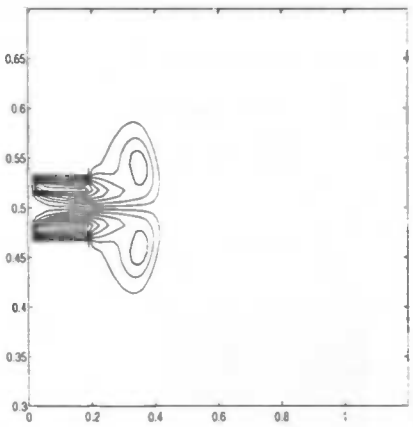


(b) t=1.00

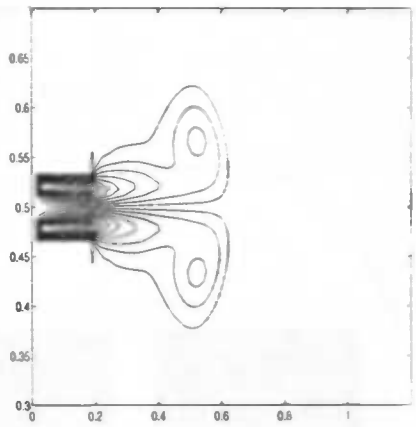


(c) t=1.52

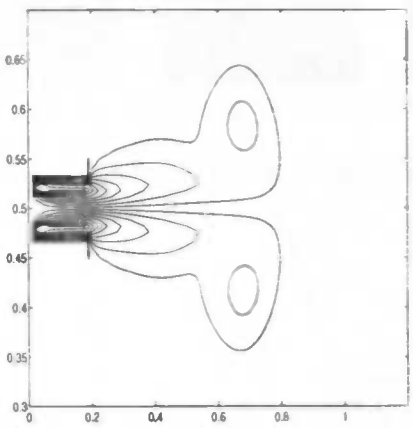
Figure 4.6: Pressure-plot along the line $x = 0$ for Re=1360



(a) t=0.48

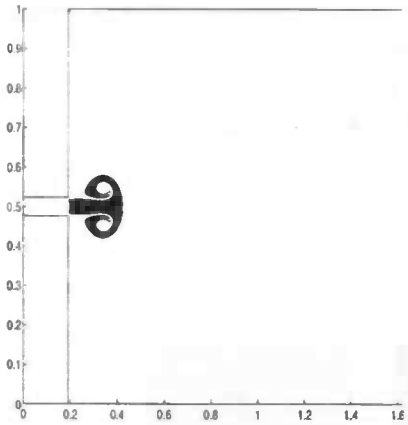


(b) t=1.00

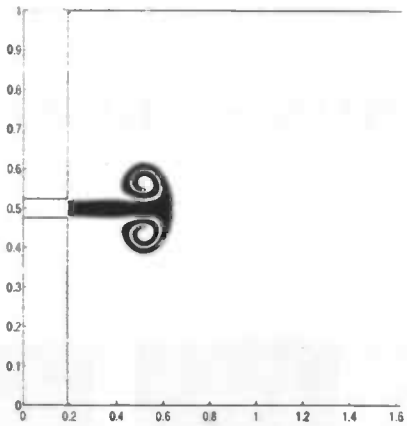


(c) t=1.52

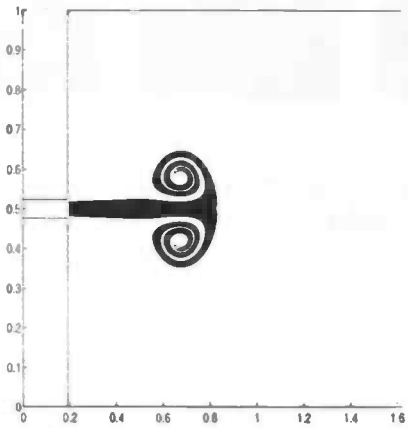
Figure 4.7: Level contourplot of the vorticity for Re=1360



(a) t=0.48



(b) t=1.00



(c) t=1.52

Figure 4.8: Streaklines of the flow for Re=1360

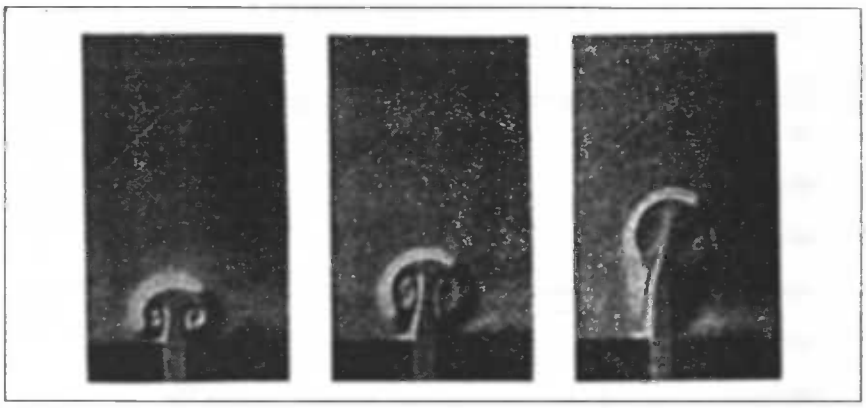


Figure 4.9: Flow visualization of the initial behavior of the starting flow from a square-edged nozzle for $Re = 1360$

Chapter 5

The Numerical Results - *case II*

5.1 Introduction

In the previous chapter we have seen how a fluid reacts on a square-edged nozzle with a steady inflow. We will now consider a time-dependent inflow with *Re* = 5127. The data for the inflow were obtained during an experiment at the University of Eindhoven and are shown in Figure 5.1.

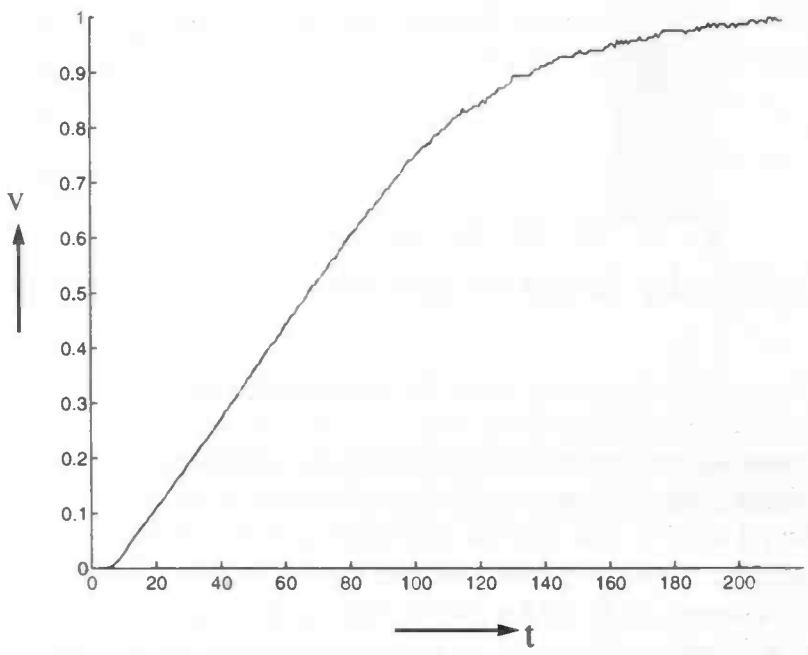


Figure 5.1: Inflow profile for case II

Near $t = 120$ we see wiggles. These wiggles originated during the experiments

in Eindhoven. This case also uses a sligthly different geometry; see Figure 5.2. Because the Reynolds number is larger than in the previous calculations, we used a 150×150 grid. As we look at Figure 5.2, we see an asterisk indicating the position of a measure point. The coordinate of this point is $(\frac{6}{76}, \frac{1}{2})$. The profile in Figure 5.1 is in fact the profile of the velocity in the center of the narrow channel. Therefore, to obtain the velocity on the line $y = 0$, we have to multiply the velocity by a factor $\frac{107}{150}$. Details on how this factor is computed, can be found in Appendix B.

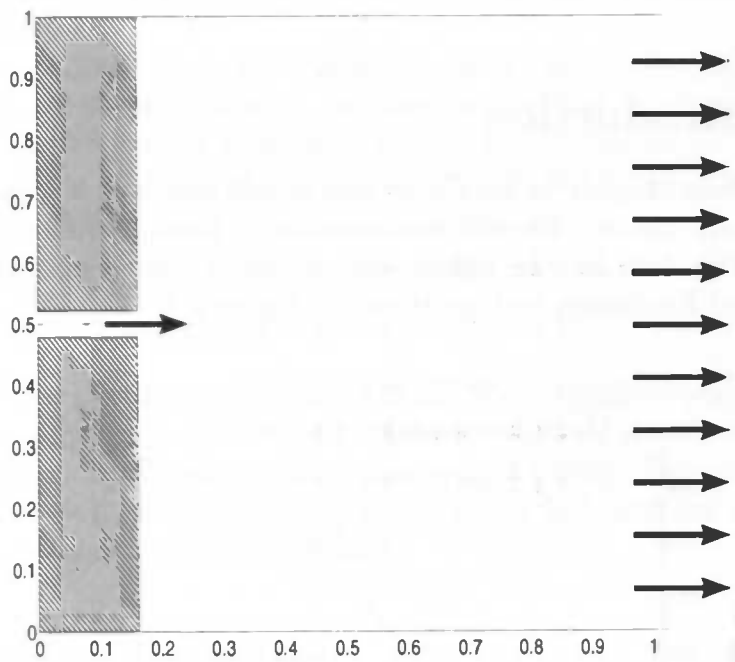


Figure 5.2: Geometry used for the computation of case II

The data cannot be implemented directly into the program, since we must find a curve that fits the data. The set is split into two parts, a curve fit for $0 \leq t \leq 7.14$ and one fit for $7.14 < t \leq 214$. For the first part we have chosen for a third degree polynomial, and for the second part a second degree polynomial. The geometry in Figure 5.2 has a length of $\frac{76}{74}$, so for $t \approx 44$, the vortices are leaving the geometry. Therefore we will only concentrate on the time interval [0,44]. Figure 5.3 shows a zoomed version of Figure 5.1 with respect to the relevant interval. When we look more closely we can see that a curve has been fitted to the data.
During one experiment pictures have been taken of the flow at five different moments, namely t=9.3, t=12.9, t=22.9, t=30.0 and t=39.3. We will discuss all

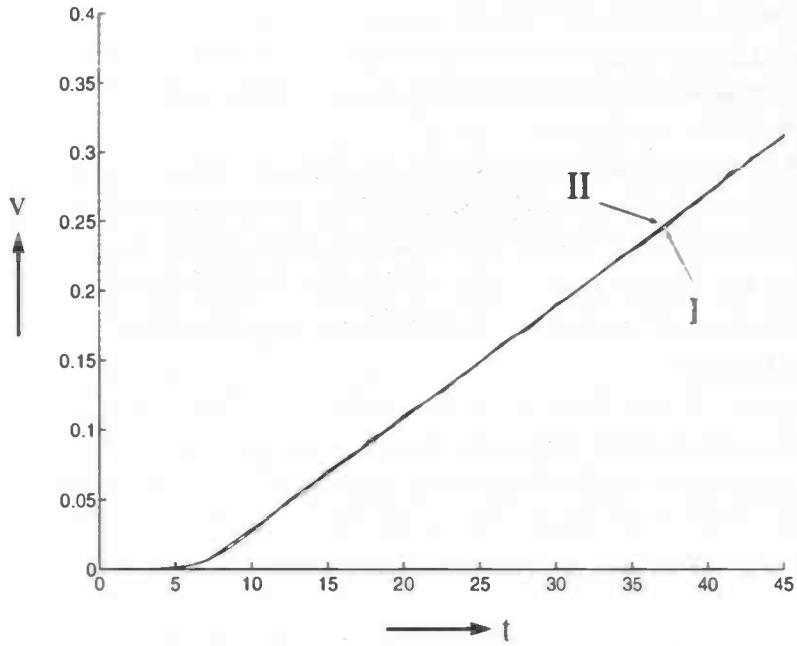


Figure 5.3: Inflow profile for t $\in$ [0,45], line I: the inflow profile curve obtained by experiments in Eindhoven, line II: curve fit for line I; for t $\in$ [0;7.1] a third order polynomial and for t $\in$ (7.1;45] a second order polynomial is fitted.

the quantities at each time-step.

5.2 Numerical results for several physical quantities

The type of pictures in this chapter is almost the same as in the previous chapter, only the Reynolds number has changed. So the discussion on the pictures, as done in chapter four, is roughly the same and we will focus only on the differences between the two cases.

5.2.1 Numerical velocity

The vector-velocity plots do not differ much from the plots in Figures 4.1, 4.2 and 4.3. Because we have used a slightly finer grid, the vector-velocity plots are finer as well. In Figures 5.4(c), 5.4(d) and 5.4(e) we can see the two vortices clearly.

5.2.2 Numerical pressure

A difference between case I and case II is that in case I no pictures were taken of the flow when the streaklines were still inside the small input channel. So the pictures of the pressure at t=9.3 and t=12.9 are slightly different. First we will discuss Figure 5.5.
Because the inflow velocity is increasing, as time passes, the pressure is increasing as well. This can be seen very clearly on the vertical axes of Figure 5.5.
When we look at Figure 5.6, we see the first differences with respect to the flow that is still inside the narrow input channel. We see that the pressure just outside the narrow input channel is positive, and becomes negative when the flow leaves the square-edged nozzle.
The plots in Figure 5.7 also show much similarity with Figure 4.6. Again Figures 5.7(a) and 5.7(b) are slightly different, because we see that the pressure along the line $x = 0$ is positive in the beginning, instead of negative as we saw in the previous chapter. This is of course due to the fact that the flow has not yet left the square-edged nozzle.

5.2.3 Numerical vorticity

In Figure 5.8 we can see the vorticity plots of the flow. Again we see concentric lines on which the vorticity is constant. In Figure 4.7 we could already distinguish two small vertical lines just next to the square-edged nozzle. Figure 5.8 shows the vertical lines as well, but larger. This is due to the fact that the viscosity is higher, resulting in larger velocity variations.

5.2.4 Numerical Streaklines

As we look at Figure 5.3, we see that at $t \approx 44$, $v \approx 0.3$. This is the maximum velocity that occurs during the simulation using the geometry in Figure 5.2. This velocity is too small for the Reynolds number 5127, which is based on $v = 1$. Only if we make the geometry longer, with $t \approx 214$ (so $v = 1$), the two vortices reach the outflow boundary and $Re = 5127$. In fact we are computing the flow with $Re = 0.3 \times 5127 = 1538$. Therefore we may expect new phenomena, when v tends to 1. In the next chapter we shall discuss this in short. For now, we must look at Figure 5.9. Again we have followed 100 points on a vertical line at each time-step. Because the inflow velocity is increasing very gently, we see that the two vortices are larger than in case I, therefore the streaklines are spreading around more. When we look at the streaklines near the coordinates $(\frac{2}{5},\frac{2}{5})$ and $(\frac{2}{5},\frac{4}{6})$, we see that these streaklines have trouble following the rotation of the flow. They tend to move backwards. As we mentioned before, this will be discussed shortly in the next chapter.

5.2.5 Experimental Streaklines

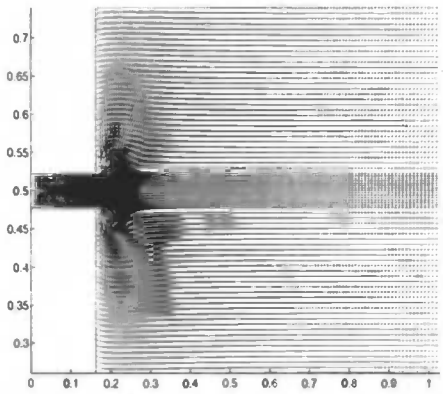
In the Figures 5.10 up to and including 5.14 we see the flow visualization of the starting flow using the inflow profile in Figure 5.3.
As stated in Section 4.2.5, the greatest differences between Figure 5.9 and Figures 5.10 up to and including 5.14 is again the incompressibility of the simulated flow. However, we can see much similarities. When we look at Figure 5.14 more closely, we can see the initial stage of turbulence. However, this turbulence can not be found in our numerical solution. Apart from this detail, the pictures in Figure 5.9 give a good representation of reality.

5.3 Calculation time

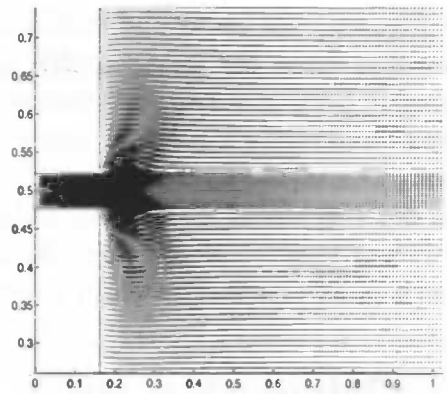
When we compare the computation time of case I to case II, we must conclude that case II demands more computer time. This is caused by several factors

1. The grid has dimensions 150×150 instead of 100×100.
2. A smaller time-step due to the finer grid.
3. The number of time-steps is approximately 8800 instead of 380.
4. The number of particles that must be followed to create the pictures in Figure 5.9 is nearly 880.000, as compared to 38.000 in case I.

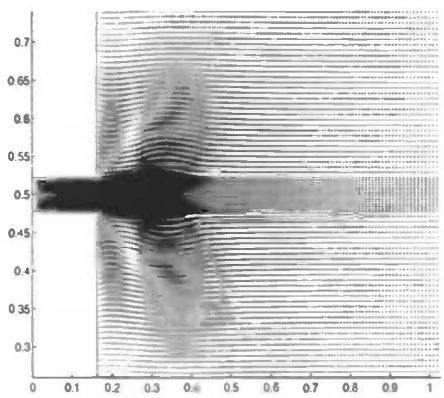
Case II can not be carried out on a Pentium-90 with 16 MB, caused by memory limitations. Therefore the computations must be done on the HP 755. A complete run will take about five days. When the DNS-program is the only process running on the HP 755, the computation can be reduced to approximately three days.



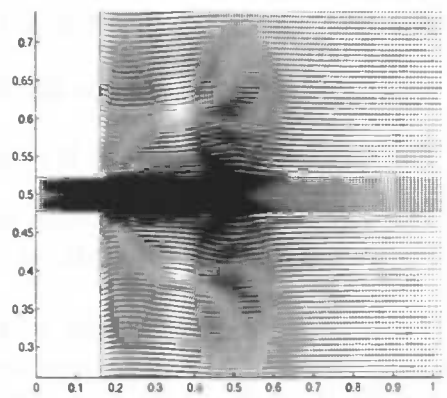
(a) t=9.3



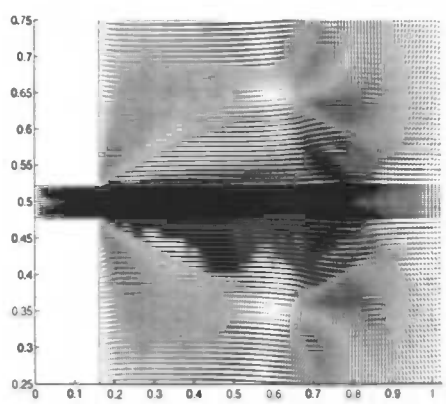
(b) t=12.9



(c) t=22.9

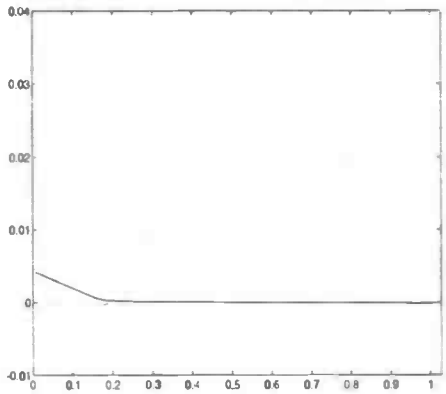


(d) t=30.0

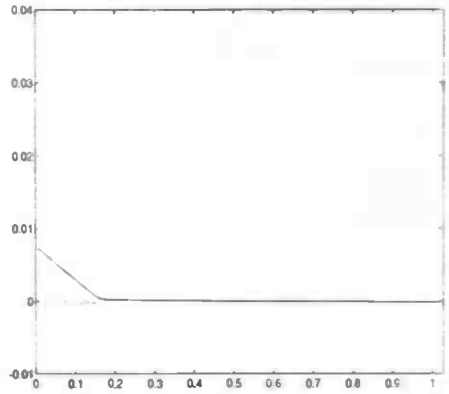


(e) t=39.3

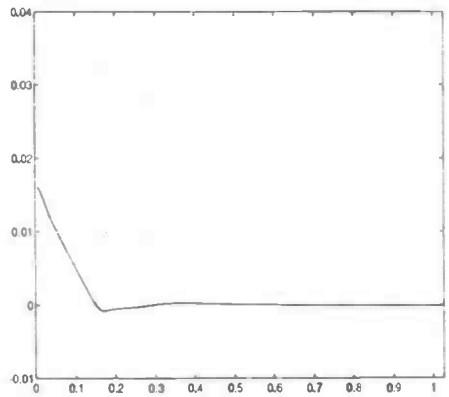
Figure 5.4: Vector-velocity plot for $Re = 5127$



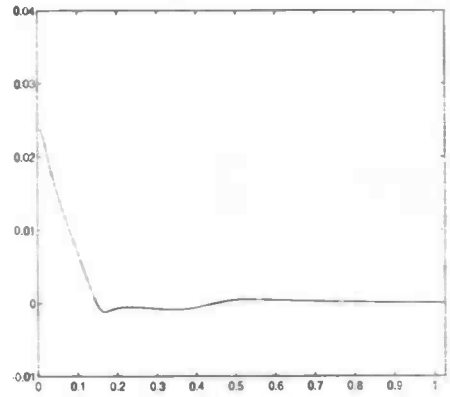
(a) t=9.3



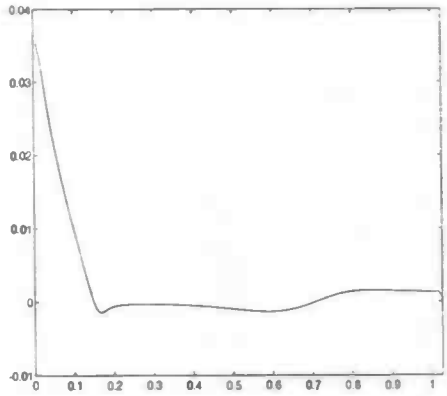
(b) t=12.9



(c) t=22.9

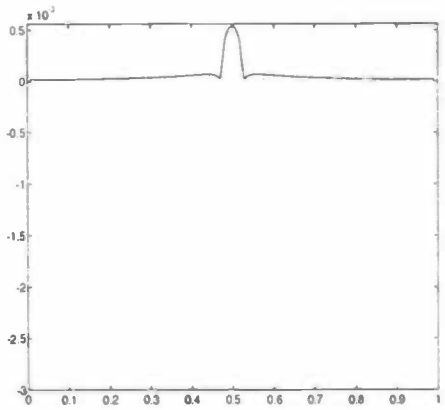


(d) t=30.0

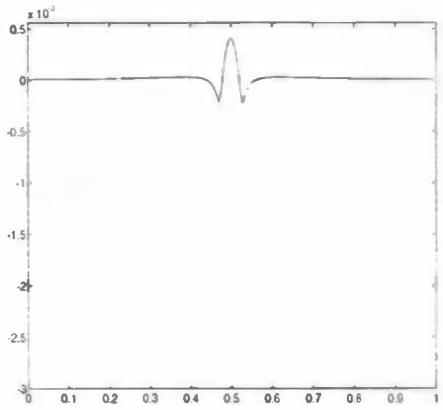


(e) t=39.3

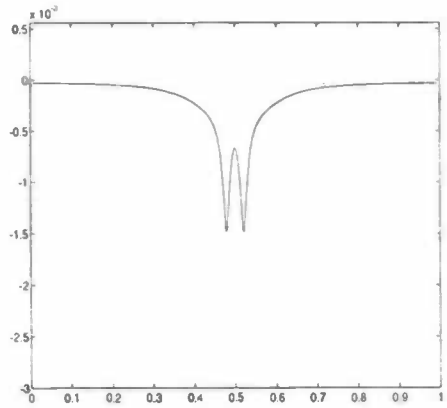
Figure 5.5: Pressure-plot along the centerline $x = \frac{1}{2}$ for $Re = 5127$



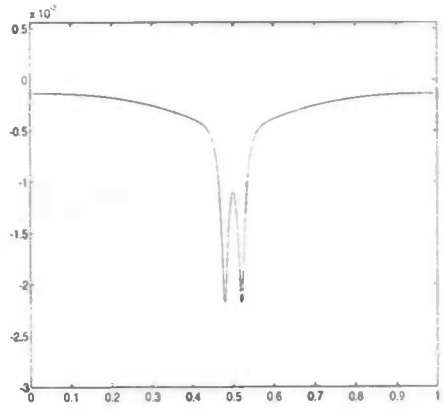
(a) t=9.3



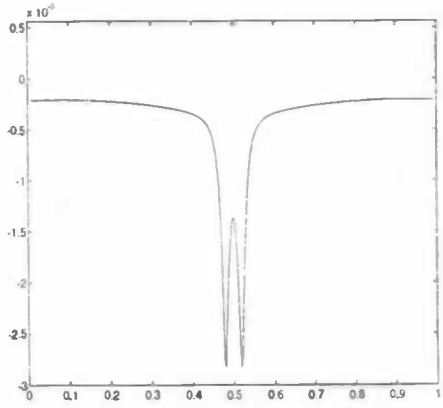
(b) t=12.9



(c) t=22.9

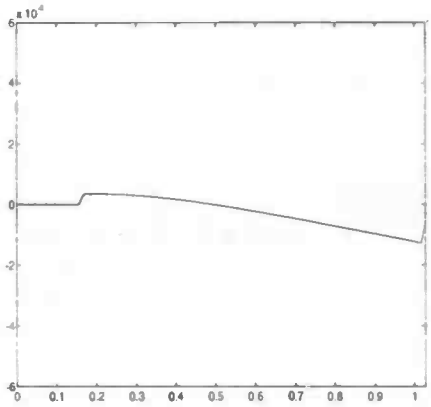


(d) t=30.0

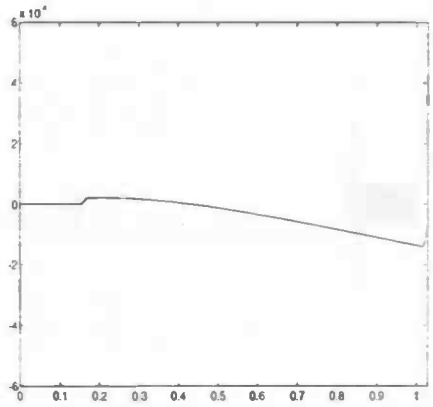


(e) t=39.3

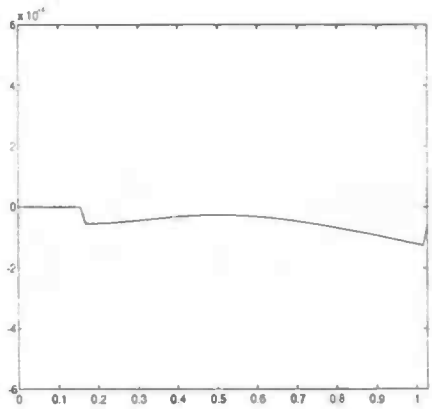
Figure 5.6: Pressure-plot along the line $y = \frac{6}{37}$ for $Re = 5127$



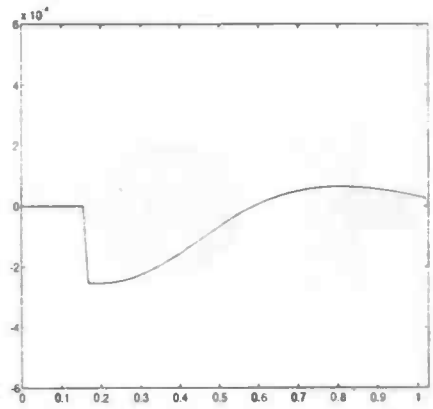
(a) t=9.3



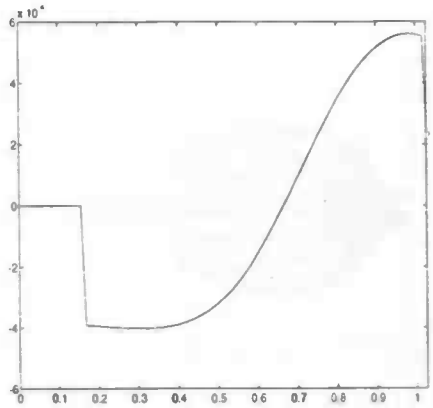
(b) t=12.9



(c) t=22.9

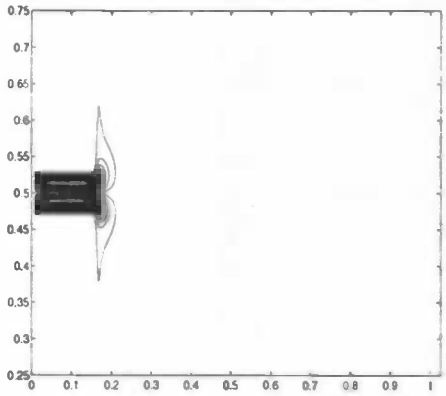


(d) t=30.0

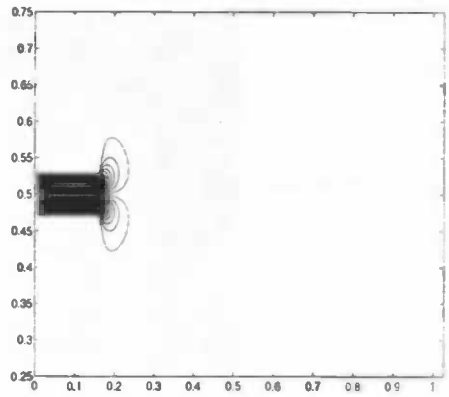


(e) t=39.3

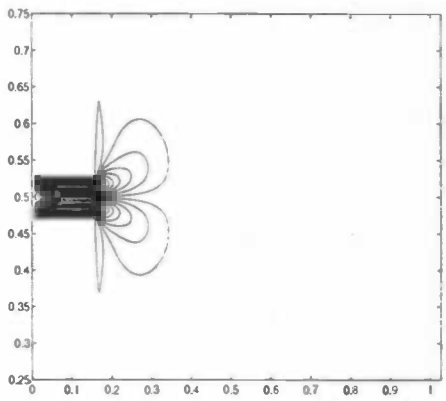
Figure 5.7: Pressure-plot along the line $x = 0$ for $Re = 5127$



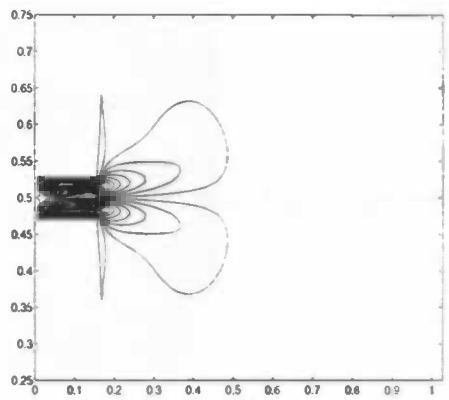
(a) t=9.3



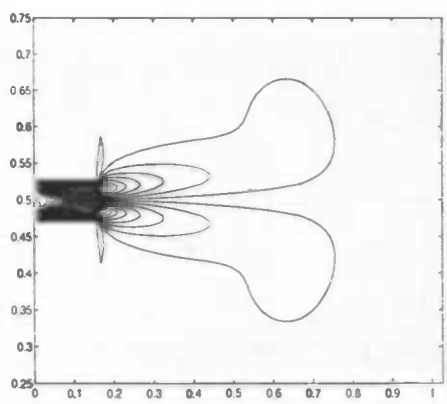
(b) t=12.9



(c) t=22.9

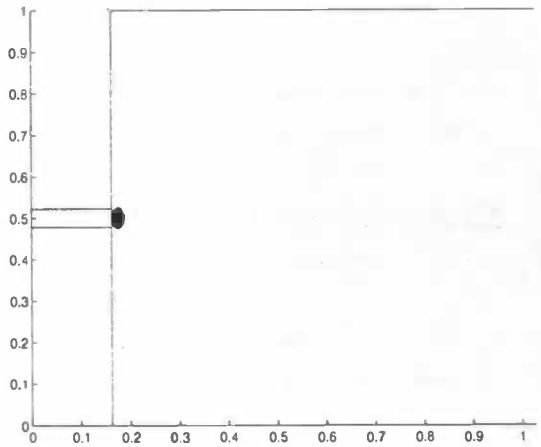


(d) t=30.0

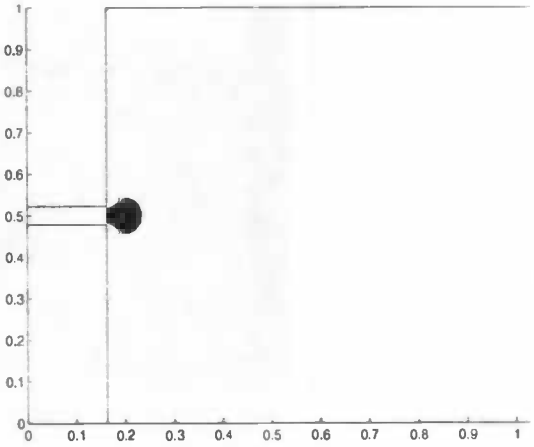


(e) t=39.3

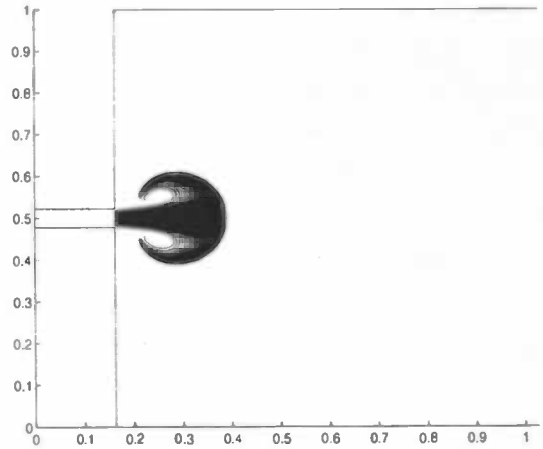
Figure 5.8: Level contourplot of the vorticity for $Re = 5127$



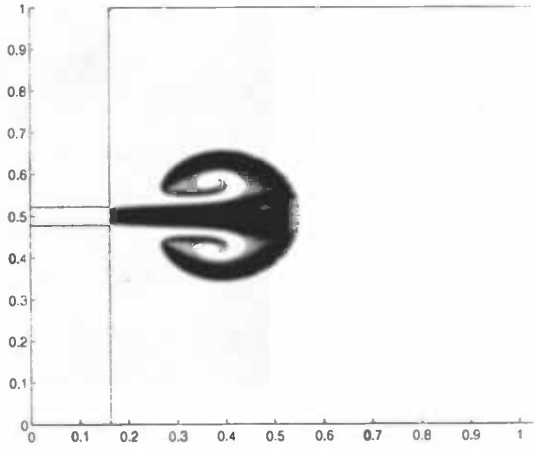
(a) t=9.3



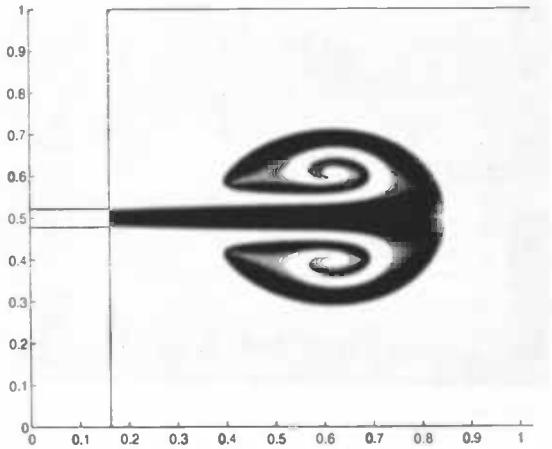
(b) t=12.9



(c) t=22.9



(d) t=30.0



(e) t=39.3

Figure 5.9: Streaklines of the flow for $Re = 5127$

Figure 5.10: t=9.3

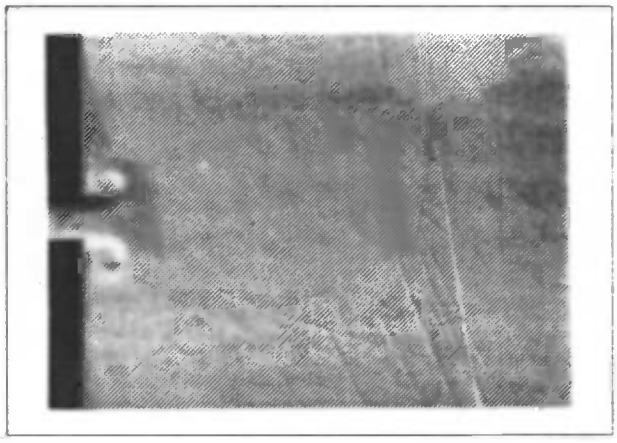


Figure 5.11: t=12.9

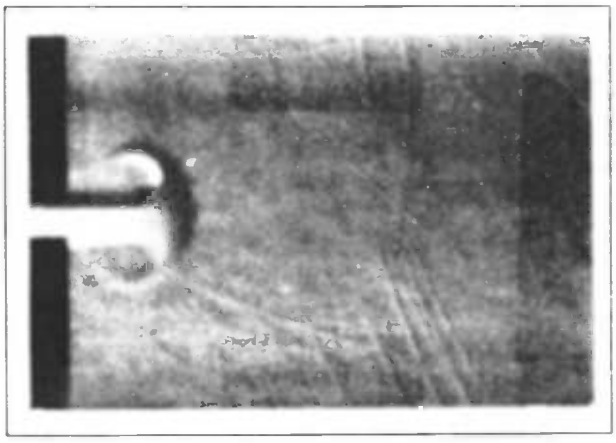


Figure 5.12: t=22.9

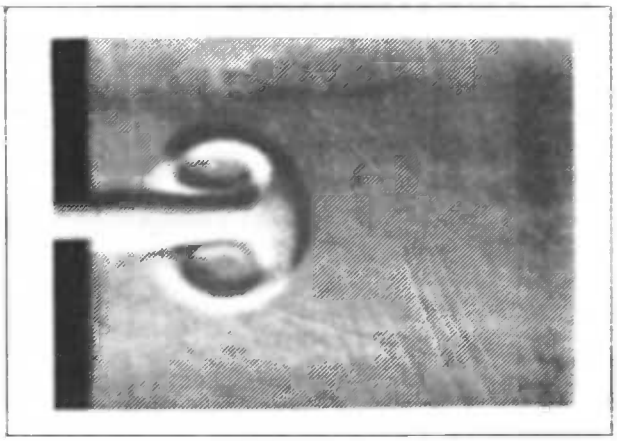


Figure 5.13: t=30.0

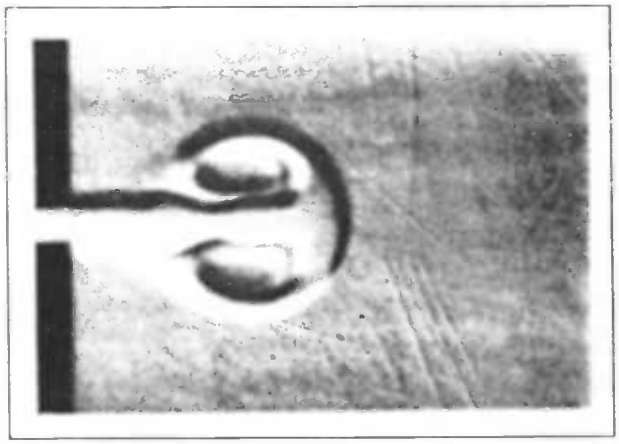


Figure 5.14: t=39.3

Chapter 6

Finishing Chapter

6.1 Conclusions

As we have seen in Figures 4.8 and 5.9, it is possible with our model to create a realistic approximation of reality. The greatest differences between numerical solution and experiments can be related with the compressibility of the flow. In Figure 5.14 we can see the initial stage of small disturbances, that cannot be found in Figure 5.9(e). Maybe a finer grid will also reveal these disturbances in the numerical solution.

6.2 Recommendations

There are still some interesting subjects left to examine more thoroughly, like

1. A different geometry, in particular the geometry in Figure 1.1(b).
2. A different inflow profile.
3. The simulation of a compressible flow.

Ad 1: to get a better model for the human vocal cords, we can choose the geometry in Figure 1.1(b). Since the human vocal cords do not have square edges, we may expect different results using a geometry with round edges.

Ad 2: we can also choose a different inflow profile. We have done very little research on this subject, but we found some interesting results. We have chosen for the inflow profile in Figure 6.1.
This inflow profile is in fact a fourth order polynomial. The Reynolds number is again 5127. The velocity however, increases much faster now. The velocity even becomes larger than one. Therefore the Reynoldsnumber during the simulation is

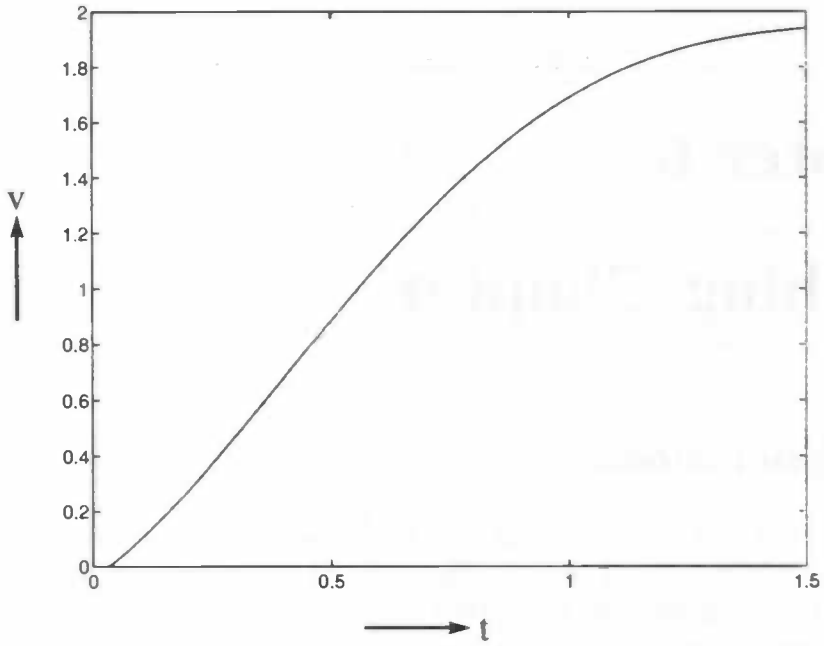


Figure 6.1: Inflow profile: fourth order polynomial

even greater than 5127. We have made a plot of the streaklines, shown in Figure 6.2.
The most interesting thing we can see in this picture is the separation of the flow. Some of the particles that are rotating in the vortices, can not follow this rotation and are separated. They move backwards, towards the square-edged nozzle.

Ad 3: when a compressible flow is simulated, we may expect that the differences between the calculations and the experiments will be smaller. However, this will demand a different simulation method since we now have to solve the following system, in vector notation

$$\left\{ \begin{array}{rcl} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) & = & 0 \\ \rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} & = & -\nabla p + \mu \Delta \mathbf{u} + \frac{1}{3}\mu \nabla(\nabla \cdot \mathbf{u}) \end{array} \right. \tag{6.1}$$

We see the quantity ρ, indicating the density. The last term has also been added to the momentum equations. This term is zero in case of an incompressible flow.

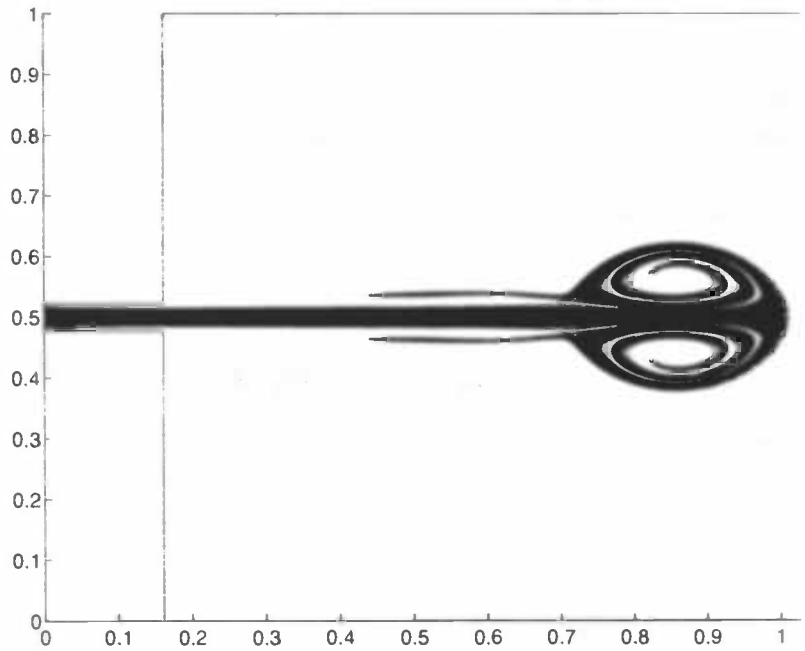


Figure 6.2: Streaklines of the flow, in which we can see separation of the flow

Appendix A

Program description

In 1993 a computer program has been designed to solve the Navier-Stokes equations in (2.5) numerically. This so-called DNS program is implemented in Fortran77. Since full documentation on the first version of the program is already available, we will only discuss the changes made for this problem. This means a discussion on the calling sequence, new variables, new subroutines and adjusted or new in- and output files.

A.1 Calling sequence

The DNS program consists of 25 subroutines: 14 subroutines for pre-processing and initialization, 8 subroutines for the time integration and 3 subroutines for post-processing. The subroutines are called in the following order:

Pre-processing/ initialization	LOGUNT	COORDN *	PRECON
	PARAMS	INFLD	DIFFUS
	CHECK	NORMP *	CONVCT
	GRID	INIMXD	INTSTR *
	INTDT *	POISSN *	
Time integration	ADJTDT *	OUTFLW *	
	BNDCND *	MAXDIV	
	INTGRT *	OUTFLD *	
	CNSTRT *		
	CNTOUR *		
Post-processing	OUTMDV		
	OUTFLD *		
	OUTCTR *		

The subroutines marked with an asterisk were adjusted or added to the program. When running the program, the pre- and post-processing routines are executed once and the time-integration is repeated during an, in advance determined, number of time-steps.

A.2 Common Block Variables

The new subroutines also use new variables, that are stored in several common blocks. Some variables were introduced for the computation of the streaklines and other subroutines, these variables were not stored in common blocks. They are:

- KN, the number of points on a vertical line that are followed during one simulation.
- KM, the number of vertical lines, with each line consisting of KN points, that are followed during one simulation. Note that KM $\leq$ NT.
- DT, the time-step at time-level n.
- ODT, the time-step at time-level n-1.
- TIME, the time at time-level n.
- OTIME, the time at time-level n-1.

/COORD/

X(I) the coordinate x_i computed from the grid.
Y(J) the coordinate y_j computed from the grid.

/CCTR/

XC(I,J,K) x-coordinate of a point on the kth line at the jth position.
YC(I,J,K) y-coordinate of a point on the kth line at the jth position.
START(I) the ith line that must be followed during a simulation.

/BNDVAL/

UY0(I) vertical velocity u on y=0.
UY1(I) vertical velocity u on y=1.
VX0(J) horizontal velocity v on x=0.
VX1(J) horizontal velocity v on x=1.
UAY(J) vertical velocity u on y=ay.
VAX(J) horizontal velocity v on x=ax.
VBX(J) horizontal velocity v on x=bx.

/TEMP/

TMP used for the name of a temporary data file.

A.3 Subroutines

As stated above we will only discuss the new subroutines in the DNS program. For every subroutine we give the input and output variables and a short description.

INTDT

Input: NX,NY,/GRID/
Output: MINDX,MINDY
Description: calculates the smallest gridwidth in the x- and y-direction.

COORDN

Input: NX,NY,/GRID/
Output: X(I),Y(J)
Description: calculates the coordinates of the gridpoints.

NORMP

Description: the average pressure over the geometry is divided by the area of the geometry without the intersections.

POISSN

Description: the pressure in the intersections is calculated in such a way that it is zero.

INTSTR

Input: KM,KN,NT,/XYCOORD/
Output: XC(I,J,K),YC(I,J,K),START(M)
Description: initialises the points that must be followed during one simulation.

ADJTDT

Input: NX,NY,MINDX,MINDY,RE,ODT,/VELOS/
Output: DT

Description: automatically adjusts the time-step.

BNDCND

Description: each time-step the boundary conditions are adjusted, resulting in a time-dependent inflow.

INTGRT

Description: integration is now done over the new geometry using the new boundary conditions, also the outflow boundary is adjusted here.

CNSTRT

Description: loops have been adjusted for the new geometry.

CNTOUR

Input: NX,NY,N,KM,DT,KN,/XYCOORD/,/GRID/,/VELOS/
Output: XC(0,J,K),YC(0,J,K)
Description: calculates the transition of a particle during one time-step.

OUTFLW

Input: NX,NY,N,KN,TIME,DT,/TEMP/,/CCTR/,/PRESS/,/VELOS/,/GRID/
Output: file CTR*xxx*.DAT, file GRD*xxx*.DAT where *xxx* is a number.
Description: periodically writes a data file to disk.

OUTFLD

Description: adjusted such that the subroutine also calculates the velocities u and v in the same gridpoint via interpolation.

OUTCTR

Input: KM,KN,/CCTR/
Output: file CONTOUR.DAT
Description: writes the data file concerning the streaklines to disk.

Appendix B

Mathematical deductions

In Chapter 5 we mentioned that the constant inflow profile must be multiplied by a factor $\frac{107}{150}$. This factor can be written as $\frac{2}{3} \times 1.07$. We shall now deduce how this factor is computed. In Figure B.1 we see a simple 2-D Poiseuile pipe.

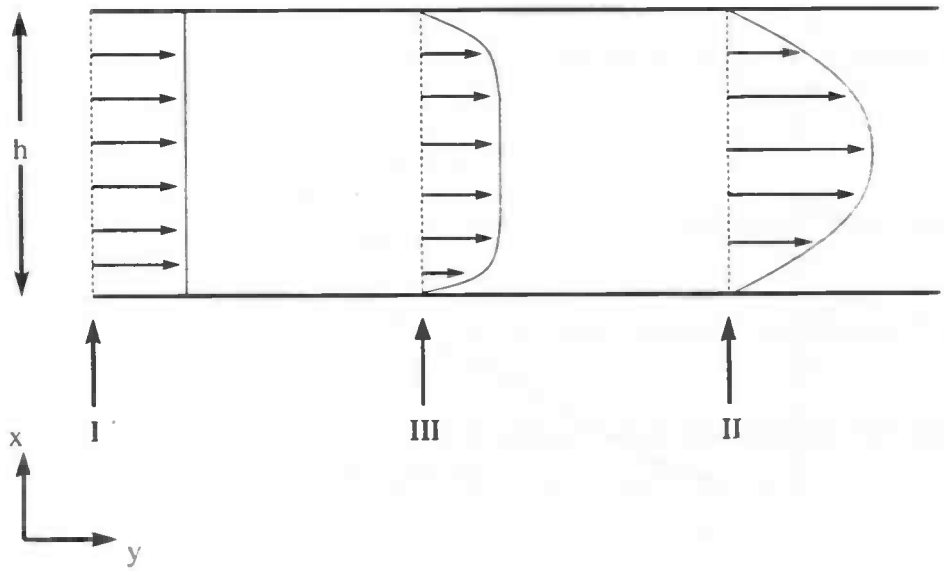


Figure B.1: Transition from constant inflow profile to Poiseuile profile

At the beginning of the pipe, marked as I, a constant inflow profile enters the pipe. This profile gradually changes into a Poiseuile profile, which is in fact a parabolic profile. This profile is drawn at position II. The maximum velocity at I is v_0 and the maxmimum velocity at II is v_1. We now have two flow profiles:

1. Profile at I: $v = v_0$
2. Profile at II: $v = v_1 \frac{4x(h-x)}{h^2}$

Using these two profiles we will deduce a relationship between v_0 and v_1. First compute the mass-flux at I:

$$\varphi_I = \int_0^h v \, dx = hv_0 \tag{B.1}$$

Now compute the mass-flux at II:

$$\varphi_{II} = \int_0^h v \, dx = \frac{4v_1}{h^2} \int_0^h x(h-x) \, dx = \frac{4v_1}{h^2} \left[\frac{1}{2}hx^2 - \frac{1}{3}x^3 \right]_0^h = \frac{2}{3}hv_1 \tag{B.2}$$

Mass-balance demands that the mass-flux at I must be equal to the mass-flux at II, i.e. $\varphi_I = \varphi_{II}$. From this equation follows a relationship between v_0 and v_1, namely $v_0 = \frac{2}{3}v_1$. This means that if we know the velocity at II we can compute the velocity at I, if and only if the velocity profile at II is described by a second order polynomial and the profile at I is constant.

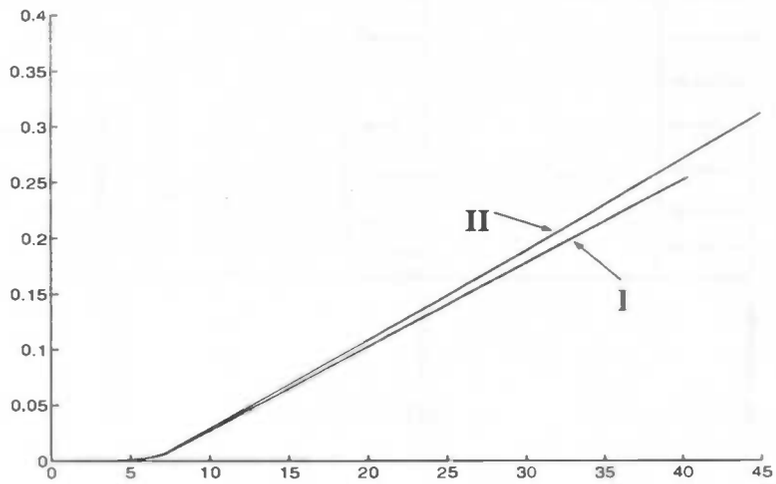


Figure B.2: Line I: velocity computed at $(\frac{6}{76}, \frac{1}{2})$, line II: velocity expected at $(\frac{6}{76}, \frac{1}{2})$

When a run was done and the velocity at $(\frac{6}{76}, \frac{1}{2})$ was computed, we found the velocity graph I in Figure B.2. However, we expected to get graph II in Figure B.2. In order to make the two graphs coincide we had to multiply graph I, and therefore the factor $\frac{2}{3}$, by 1.07. This factor was obtained by experimenting with the two graphs, until they coincided. A reason why the muliplication factor is not $\frac{2}{3}$ but $\frac{107}{150}$, is that at $(\frac{6}{76}, \frac{1}{2})$ a Poiseuile profile had not yet arised. The shape of this profile can be found in Figure B.1 at III.

Bibliography

[1] T.M. Apostol. *Calculus*, volume II. Wiley, 1969.

[2] Kendall E. Atkinson. *An Introduction to Numerical Analysis.* Wiley, 1989.

[3] E.F.F. Botta. *Eindige differentiemethoden.* Rijksuniversiteit Groningen, November 1992. Lecture notes (in Dutch).

[4] H.W. Hoogstraten. *Stromingsleer.* Rijksuniversiteit Groningen, June 1992. Lecture notes (in Dutch).

[5] M.C.A.M. Peters. *Aeroacoustic Sources in Internal Flows.* PhD thesis, University of Eindhoven, 1993.

[6] Peyret & Taylor. Incompressible flows.

[7] E. Thomas. *Integraalrekening.* Rijksuniversiteit Groningen, 1994. Lecture notes (in Dutch).

[8] A.E.P. Veldman. *Grenslaagstromingen.* Rijksuniversiteit Groningen, December 1992. Lecture notes (in Dutch).

[9] A.E.P. Veldman. *Numerieke Stromingsleer.* Rijksuniversiteit Groningen, March 1994. Lecture notes (in Dutch).

List of Figures