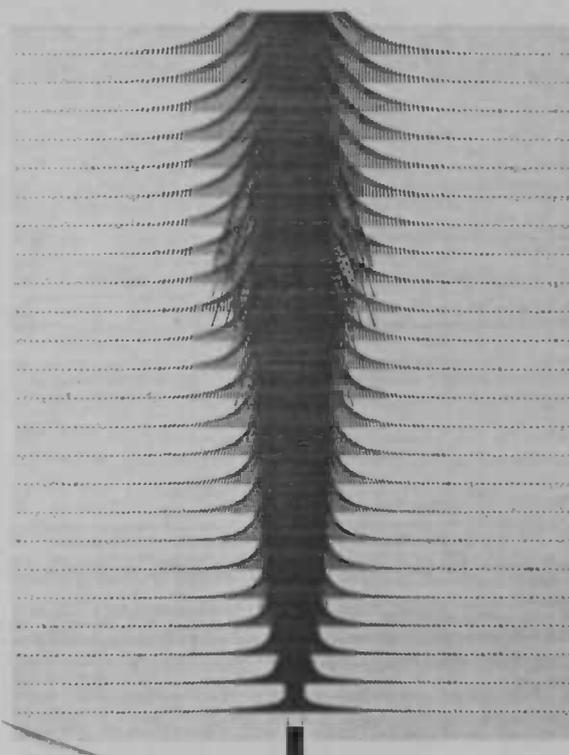


WORDT  
NIET UITGELEEND



# Numerical Simulation of a Turbulent Jet

Jan Mente Drent



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

Department of  
Mathematics

RUG



Master's thesis

---

# Numerical Simulation of a Turbulent Jet

Jan Mente Drent

---

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

March 1998

# Preface

This thesis is a report of my graduation work at the mathematics department of the University of Groningen. I started working on the assignment last July. The assignment has been done to support the PhD. research of Ir. Roel Luppens at the Technical University of Eindhoven: Numerical Solution Methods and Turbulence Models for Turbulent Jets and Diffusion Flames. His advises and of course those of Prof. Dr. A.E.P. Veldman have lead to this final report. Therefore I would like to thank them both. Further I would like to thank fellow graduate student Bernard Bos for his practical help on several things. Finally I have to thank my girlfriend Mirjam Deinum, who always kept faith in me during this long period.

The report has become an attempt to describe a turbulent jet. With Prandtl's mixing-length model the problem of turbulence has been taken care of, and with MILU an efficient solving method has been used. These features should make the simulation of a turbulent jet by HEAT97 possible.

I hope that the reader has a pleasant time, studying this report.

Jan Mente Drent, Groningen  
March 1998

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Mathematical Model</b>	<b>4</b>
2.1	2D laminar flow . . . . .	4
2.2	Axisymmetrical laminar flow . . . . .	5
2.2.1	Navier-Stokes . . . . .	5
2.2.2	Analytic solutions of a laminar jet . . . . .	6
2.3	Axisymmetrical turbulent flow . . . . .	8
2.3.1	Navier-Stokes . . . . .	9
2.3.2	Analytic solutions of a turbulent jet . . . . .	9
<b>3</b>	<b>Numerical Model</b>	<b>10</b>
3.1	Discretization and solving . . . . .	10
3.1.1	Geometry . . . . .	10
3.1.2	The Poisson Equation . . . . .	10
3.1.3	Spatial Discretization . . . . .	11
3.1.4	Iteration - MILU . . . . .	12
3.1.5	Boundary, in- and outflow conditions . . . . .	13
3.2	Turbulent extension of HEAT97 . . . . .	13
<b>4</b>	<b>Results</b>	<b>15</b>
4.1	Testing HEAT97 . . . . .	15
4.2	The laminar jet . . . . .	17
4.3	The turbulent jet . . . . .	21
<b>5</b>	<b>Conclusions and recommendations</b>	<b>25</b>
<b>A</b>	<b>Program description</b>	<b>26</b>
A.1	FORTRAN77 . . . . .	26
A.1.1	Calling sequence . . . . .	26
A.1.2	Common block variables . . . . .	26
A.1.3	Subroutines . . . . .	28
A.1.4	HALFW . . . . .	29
A.2	Input files . . . . .	30
A.2.1	Main input . . . . .	30
A.2.2	Weird geometry . . . . .	33
A.3	Output files . . . . .	34

# Chapter 1

## Introduction

In a time where the environment is very important to us, all possible ways of reducing the use of fuel are regarded. This can be done by trying to find other energy sources, but as long as there's no total replacing alternative it's useful e.g. to optimize the existing (turbulent) combustion processes. In the industry many of those turbulent combustion processes occur, like in huge heating turbines, furnaces and ovens.

At the Technical University of Eindhoven (TUE) an effort is made to get a grip on turbulent combustion. Ir. Roel Luppens is trying to do so in his PhD research 'Numerical Solution Methods and Turbulence Models for Turbulent Jets and Diffusion Flames'. To that the turbulent flames have been modeled according to the flamelet principle. This means that the chemical equations, which are part of the description of the turbulent flame, are separated from the other flow-equations, and are solved a priori. (This separation is necessary because the solving of the total system isn't possible up-till now.) The solutions of these chemical equations are stored in tables, the so called Flamelet Library, in which every table corresponds to a 1D flamelet. These tables are used iteratively when the turbulent flow pattern is computed. This means that the chemistry is combined to the flow pattern and in this way the flame is build up with 1D flamelets.

The turbulent flow pattern is the foundation to the computation of the flame. Therefore it's necessary to model the jet flow accurately. Important to this are the turbulence model, and the numerical computation of the transport equations. In Eindhoven this is done by TEACHT, a code that is using the 'hybrid-scheme' for discretization and a variant of the SIMPLE method as solver. This makes the solution sometimes very inaccurate, and the convergence is very slow. Therefore another program might give better results.

This report is an attempt to get more accurate results with less time used. With HEAT97, an upgraded version of SAVOF96 [1], the turbulent jet-stream is simulated using Prandtl's mixing-length model for the turbulence. The discretization used by HEAT97 is full-upwind, which gives also first order accuracy. The solver used is a variant to a 2D-MICCG method. The last feature has to result in the desired improvements.

To get a clear view of what has been done, the mathematics, the numerical properties and the obtained results have been written in different chapters. For a short description of HEAT97 the reader is referred to the Appendix.

## Chapter 2

# Mathematical Model

In this chapter the mathematical assumptions are described. All the used variables are clarified in the 'List of symbols' at the end of this report. Most of the symbols are commonly used or clear to the reader. Unclear variables will be defined when they appear.

### 2.1 2D laminar flow

Fluid motion is dictated by the incompressible Navier-Stokes equations. Those are deduced from two mechanical conservation laws. Below those are written (for 2D flows) in mathematical equations.

*Conservation of mass:*

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0$$

*Conservation of momentum:*

$$\begin{aligned} \frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} &= \rho F_x + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \\ \frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv)}{\partial x} + \frac{\partial (\rho v^2)}{\partial y} &= \rho F_y + \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \end{aligned}$$

$F_x$  and  $F_y$  are the components of an external force in  $x$ - and  $y$ -direction respectively, and  $\sigma = (\sigma_{ij})$  is the stress tensor. For incompressible Newtonian fluids, it's known that the stress is linearly related to the rate of deformation. This is the case we study here, so we take

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} = \begin{pmatrix} -p & 0 \\ 0 & -p \end{pmatrix} + \mu \begin{pmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} \end{pmatrix}$$

Using this definition of  $\sigma$ , the 2D incompressible Navier-Stokes equations to be solved become:

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

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

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

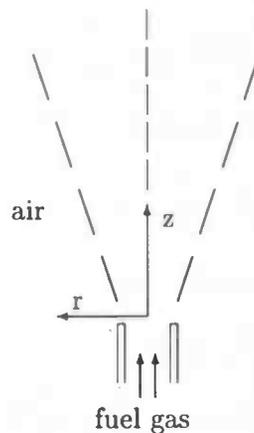
where the kinematic viscosity  $\nu = \mu/\rho$  has been introduced. For later use it's convenient to write the Navier-Stokes equations in their vector notation:

$$\begin{aligned} \operatorname{div} \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \operatorname{grad}) \mathbf{u} &= \mathbf{F} - \frac{1}{\rho} \operatorname{grad} p + \nu \operatorname{div} \operatorname{grad} \mathbf{u} \end{aligned}$$

The boundary conditions for viscous fluids are as usual chosen as  $\mathbf{u} = 0$  on a solid boundary. This describes two phenomena: impermeability of the solid wall and the fact that the fluid 'sticks' to the wall due to its viscosity. The latter is often referred to as the *no-slip* condition.

## 2.2 Axisymmetrical laminar flow

The situation to be considered (from now on) is represented by the sketch below:



From below fuel gas is injected through a small axisymmetric pipe into the air. There it mixes with air and spreads out from the injection plain. This describes the dynamics of the jet shortly.

The whole problem is considered as being axisymmetric. Therefore the symmetry axis  $z$  and the radial axis  $r$  are the space variables to be regarded.

### 2.2.1 Navier-Stokes

The equations used by HEAT97 are (2.1)-(2.3) but then transformed with cylindrical coordinates  $(r, z, \varphi)$ . Although in the axisymmetrical case  $(\partial/\partial\varphi = 0)$  one needs an extra direction (*azimuthal direction*) it's not necessary for the jet, because the flow is assumed to have no swirl ( $w = 0$ ). So the equations used for the jet are:

$$\frac{1}{r} \frac{\partial (ru)}{\partial r} + \frac{\partial v}{\partial z} = 0 \quad (2.4)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + v \frac{\partial u}{\partial z} = F_r - \frac{1}{\rho} \frac{\partial p}{\partial r} + \nu \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{\partial^2 u}{\partial z^2} - \frac{u}{r^2} \right) \quad (2.5)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial r} + v \frac{\partial v}{\partial z} = F_z - \frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial v}{\partial r} \right) + \frac{\partial^2 v}{\partial z^2} \right) \quad (2.6)$$

### 2.2.2 Analytic solutions of a laminar jet

The jet problem is described analytically by Spalding [2]. In his description of the jet the following properties of the physics are assumed: the surrounding air is at rest; the density, viscosity and other properties of the gases are uniform; the flow is steady; the pressure in the flow is uniform; and the diffusion and viscous action in the axial direction are negligible.

With these assumptions the mathematical description becomes:

*Mass conservation:*

$$\frac{\partial}{\partial z} (\rho v r) + \frac{\partial}{\partial r} (\rho u r) = 0$$

*Momentum conservation in axial direction:*

$$\frac{\partial}{\partial z} (\rho v r v) + \frac{\partial}{\partial r} (\rho u r v) = \frac{\partial}{\partial r} \left( \mu r \frac{\partial v}{\partial r} \right)$$

Since  $\rho$  and  $\mu$  are being taken as uniform it's possible to define:

$$\nu \equiv \frac{\mu}{\rho}$$

Rewriting the equations results in:

$$\frac{\partial v}{\partial z} + \frac{\partial u}{\partial r} + \frac{u}{r} = 0 \quad (2.7)$$

$$v \frac{\partial v}{\partial z} + u \frac{\partial v}{\partial r} = \frac{\nu}{r} \frac{\partial}{\partial r} \left( r \frac{\partial v}{\partial r} \right) \quad (2.8)$$

A short remark must be made to these equations. Compared to the 2D incompressible Navier-Stokes equations some differences occur. For example in this description of the jet the pressure isn't variable. Also the diffusive term in axial direction is disregarded.

The boundary conditions to this problem are:

*At large radius:*

$$r \rightarrow \infty : v = 0 \quad (2.9)$$

*At the entrance plane:*

$$z = 0, \quad r \leq r_0 : v = v_0 \quad (2.10)$$

$$z = 0, \quad r > r_0 : v = 0 \quad (2.11)$$

with  $r_0$  and  $v_0$  being the radius of the inflow pipe and the inflow velocity respectively. To solve these equations (for large values of  $z$ ) we will first define a jet-invariant.

$$\begin{aligned} I_v &\equiv \frac{1}{\nu} \int_0^{\infty} v^2 r \, dr \\ &= \frac{1}{\nu} \left( \frac{1}{2} v_0^2 r_0^2 \right) \end{aligned} \quad (2.12)$$

This quantity is equal to  $1/(\pi\mu)$  times the rate of flow of axial momentum.

It can be verified by substitution that the relations below satisfy the differential equations (2.7) and (2.8), and the boundary condition (2.9). They don't satisfy the boundary conditions for the injection plane ( $z = 0$ ) in detail; but they agree with these conditions in respect of the integral relation (2.12). They can therefore be accepted as solutions which are valid for large values of  $z$ , at which the effects of the details of the profile of  $v$  at  $z = 0$  have become insignificant. The equations are:

$$\begin{aligned} u &= \left( \frac{3}{8} I_v \nu \right)^{\frac{1}{2}} \frac{\xi}{z} \left( 1 - \frac{\xi^2}{4} \right) \left( 1 + \frac{\xi^2}{4} \right)^{-2} \\ v &= \frac{3}{4} \frac{I_v}{z} \left( 1 + \frac{\xi^2}{4} \right)^{-2} \end{aligned}$$

where the single non-dimensional space variable  $\xi$  is defined by:

$$\xi \equiv \left( \frac{3}{8} \frac{I_v}{\nu} \right)^{\frac{1}{2}} \frac{r}{z}$$

### Discussion

Since  $\xi$  is constant for fixed  $r/z$ , the jet can be regarded as radiating from the injection point in a conical manner.

The value of  $v$  on the axis ( $\xi = 0$ ) varies reciprocally with longitudinal distance  $z$ , in accordance with

$$v_{ax} z = \frac{3}{4} I_v \quad (2.13)$$

Inspection of this formula confirms that the solution can't be valid at small  $z$ ; for  $v_{ax}$  can't exceed the entrance velocity  $v_0$ . It follows that the solutions are valid only for the condition:

$$z > \frac{3}{4} \frac{I_v}{v_0} \quad \Leftrightarrow \quad \frac{z}{r_0} > \frac{3}{8} \frac{v_0 r_0}{\nu}$$

This formula indicates that the number of nozzle radii downstream at which the injection details are submerged is of the order of the Reynolds number based on the nozzle conditions. The radial profile of  $v$ , when normalized by its axial velocity, is:

$$\frac{v}{v_{ax}} = \frac{1}{\left( 1 + \frac{\xi^2}{4} \right)^2}$$

The sketch in figure 2.1 (see next page) represents the shape qualitatively. The significance of the value  $\xi = 1.287$  is that it's often useful to characterize the width of a jet by the value

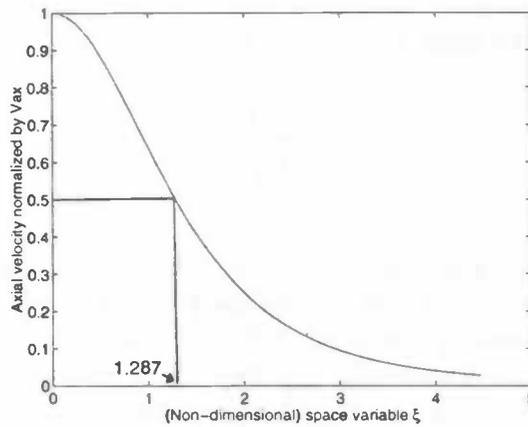


Figure 2.1:

of  $r_{\frac{1}{2}}$ , i.e. the radius at which the velocity has one half of its velocity on the symmetry axis. This leads to a property for the laminar jet, namely:

$$\begin{aligned}
 \frac{r_{\frac{1}{2}}}{z} &= 1.287 \left( \frac{8\nu}{3I_v} \right)^{\frac{1}{2}} \\
 &= 1.287 \left( \frac{16}{3} \right)^{\frac{1}{2}} \left( \frac{\nu}{v_0 r_0} \right) \\
 &= 2.97 \left( \frac{\nu}{v_0 r_0} \right)
 \end{aligned} \tag{2.14}$$

This shows that the angle of the laminar jet is inversely proportional to the Reynolds number.

### 2.3 Axisymmetrical turbulent flow

Because turbulent flow is difficult to get a grip on, it has to be modeled with rough assumptions. In this study mixing-length models will be used. Therefore the viscosity will be written as (used for the analytical solution of the turbulent jet)

$$\begin{aligned}
 \mu_{eff} &\approx 0.0102 \rho |v_{ax}| \delta \\
 &\approx 0.0255 \rho |v_{ax}| r_{\frac{1}{2}}
 \end{aligned} \tag{2.15}$$

or as (used by HEAT97)

$$\mu_{eff} \approx \rho l_m^2 \left| \frac{\partial v}{\partial r} \right| \tag{2.16}$$

The 'thickness' of a mixing-layer can't be determined unequivocally. For a jet  $r_{\frac{1}{2}}$  can be determined. Therefore the following assumptions are made to express  $\delta$  and  $l_m$ : the jet width  $\delta$  is approximately  $2.5 r_{\frac{1}{2}}$ , and the mixing-length  $l_m \approx 0.075 \delta \approx 0.1875 r_{\frac{1}{2}}$ , because the profiles of velocity in a turbulent jet are fairly uniform.

### 2.3.1 Navier-Stokes

Because we're using Prandtl's mixing-length model (2.16) the viscosity is no longer place independent. Therefore the equations (2.4)-(2.6) have to be changed a little. In a turbulent jet the equations are:

$$\begin{aligned} \frac{1}{r} \frac{\partial (ru)}{\partial r} + \frac{\partial v}{\partial z} &= 0 \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + v \frac{\partial u}{\partial z} &= F_r - \frac{1}{\rho} \frac{\partial p}{\partial r} + \frac{\partial}{\partial r} \left( \nu \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial u}{\partial z} \right) + \frac{\nu}{r} \left( \frac{\partial u}{\partial r} - \frac{u}{r} \right) \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial r} + v \frac{\partial v}{\partial z} &= F_z - \frac{1}{\rho} \frac{\partial p}{\partial z} + \frac{\partial}{\partial r} \left( \nu \frac{\partial v}{\partial r} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial v}{\partial z} \right) + \frac{\nu}{r} \frac{\partial v}{\partial r} \end{aligned}$$

### 2.3.2 Analytic solutions of a turbulent jet

The situation is precisely as for the laminar jet, except that the effective viscosity of equation (2.15) replaces the laminar viscosity.

The axial velocity can be predicted with the given relations. From equations (2.14) and (2.15), we have:

$$\begin{aligned} \frac{r_{\frac{1}{2}}}{z} &= 2.97 \left( 0.0255 \frac{v_{ax} r_{\frac{1}{2}}}{v_0 r_0} \right) \\ \Rightarrow \frac{v_{ax}}{v_0} &= 13.2 \frac{r_0}{z} \\ &= 6.6 \frac{D_0}{z} \end{aligned} \tag{2.17}$$

(with  $D_0 = 2r_0$ ). Also the jet angle can be calculated. From equations (2.12) and (2.15), there results:

$$I_v = \frac{1}{2} \frac{v_0^2 r_0^2}{0.0255 v_{ax} r_{\frac{1}{2}}}$$

Combination with (2.13) yields:

$$v_{ax} z = \frac{3}{8} \frac{v_0^2 r_0^2}{0.0255 v_{ax} r_{\frac{1}{2}}} \Rightarrow \frac{v_{ax}^2}{v_0^2} = 14.7 \frac{r_0^2}{r_{\frac{1}{2}}^2 z}$$

Finally, elimination of  $r_0$  between this equation and (2.17) leads to the result:

$$\frac{r_{\frac{1}{2}}}{z} = 0.0844 \tag{2.18}$$

where the jet angle of the turbulent jet is *independent* of the Reynolds number.

## Chapter 3

# Numerical Model

In this chapter the numerical model used by HEAT97 for simulating the jet stream is discussed. The changes that were necessary to solve the turbulent jet with HEAT97, are described in the second section.

### 3.1 Discretization and solving

#### 3.1.1 Geometry

To describe e.g. the jet's geometry, HEAT97 uses an input file in which the geometry parameters (and other parameters) must be defined. Also an extra input file can be used to create complex geometries.<sup>1</sup>

The field that has been created by the input file contains cells, which have been labeled by the following values:

- 0 : full cell,
- 7 : 'outflow' cell,
- 8 : 'inflow' cell,
- 9 : boundary cell.

With this labeling one knows which equation/condition must be applied at a certain cell.

#### 3.1.2 The Poisson Equation

HEAT97 computes the pressure by means of a Poisson equation (3.3) that follows from the incompressible version of the Navier-Stokes equations. If we use the abbreviation

$$\mathbf{R} = -(\mathbf{u} \cdot \text{grad}) \mathbf{u} + \nu \text{div grad } \mathbf{u} + \mathbf{F},$$

and set  $\rho = 1$ , we can write the Navier-Stokes equations as:

$$\begin{aligned} \text{div } \mathbf{u} &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + \text{grad } p &= \mathbf{R}. \end{aligned}$$

---

<sup>1</sup>For a description of the input files, see appendix A 2

The  $\partial u/\partial t$  term is discretized in time with a forward Euler method, yielding

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

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

$\delta t$  denotes the time step (which will be reduced automatically when the CFL limit or the diffusive limit is reached),  $n$  denotes the 'old' time level and  $n + 1$  the 'new' one. Rearranging terms gives:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \delta t \mathbf{R}^n - \delta t \operatorname{grad} p^{n+1}.$$

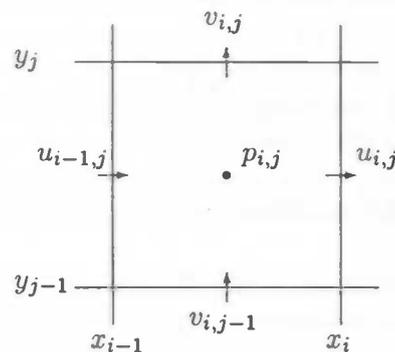
Substituting this into (3.1), we obtain

$$\operatorname{div} \operatorname{grad} p^{n+1} = \operatorname{div} \left( \frac{\mathbf{u}^n}{\delta t} + \mathbf{R}^n \right). \quad (3.3)$$

This is called the *Poisson equation* for the pressure.

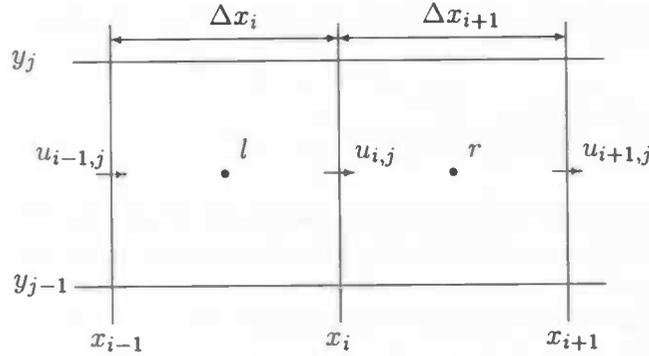
### 3.1.3 Spatial Discretization

HEAT97 uses a Cartesian grid for its discretization, with optional refinements near user specified locations. The unknown variables  $u$ ,  $v$  and  $p$  are placed as follows: the horizontal velocity  $u$  on the vertical faces of a cell, the vertical velocity  $v$  on the horizontal faces, and the pressure  $p$  in the center. This is the well known MAC (Marker-And-Cell) method. It has an important benefit compared to other methods of placement that suffer problems regarding the uniqueness of the pressure.



The Poisson equation is solved in two steps. First, in subroutine TILDE<sup>2</sup>, the momentum equation is integrated, i.e. the part  $\frac{\mathbf{u}^n}{\delta t} + \mathbf{R}^n$  of (3.3) is calculated. We will show how this is done for the  $x$ -direction. First the viscous terms in  $\mathbf{R}^n$ :  $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ .

<sup>2</sup>For a global description of the subroutine structure of HEAT97, see appendix A1



The second order derivative is discretized centrally. First:

$$\left. \frac{\partial u_{i,j}^n}{\partial x} \right|_l = \frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x_i}$$

$$\left. \frac{\partial u_{i,j}^n}{\partial x} \right|_r = \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x_{i+1}}$$

Then these are used to form

$$\frac{\partial^2 u_{i,j}^n}{\partial x^2} = \frac{\left. \frac{\partial u_{i,j}^n}{\partial x} \right|_r - \left. \frac{\partial u_{i,j}^n}{\partial x} \right|_l}{\frac{1}{2}(\Delta x_i + \Delta x_{i+1})}$$

That takes care of the diffusive terms, now the convective terms  $u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y}$ . These are treated with upwind discretization, which is controlled by the upwind parameter  $\alpha$  ( $\alpha = 1$  is full upwind).

$$u_{i,j}^n \frac{\partial u_{i,j}^n}{\partial x} = \begin{cases} \frac{u}{(1+\alpha)\Delta x_i + (1-\alpha)\Delta x_{i+1}} \left( (1-\alpha)\Delta x_{i+1} \left. \frac{\partial u_{i,j}^n}{\partial x} \right|_r + (1+\alpha)\Delta x_i \left. \frac{\partial u_{i,j}^n}{\partial x} \right|_l \right), & u \geq 0 \\ \frac{u}{(1-\alpha)\Delta x_i + (1+\alpha)\Delta x_{i+1}} \left( (1+\alpha)\Delta x_{i+1} \left. \frac{\partial u_{i,j}^n}{\partial x} \right|_r + (1-\alpha)\Delta x_i \left. \frac{\partial u_{i,j}^n}{\partial x} \right|_l \right), & u < 0 \end{cases}$$

If the same is done for the  $y$ -direction and for  $v$ , the Poisson equation can be solved by an iterative process as described in the next section.

### 3.1.4 Iteration - MILU

For presentational reasons, let's write the Poisson equation like

$$Ax = b,$$

where

$$A = \text{div grad}, \quad x = p^{n+1}, \quad b = \text{div} \left( \frac{u^n}{\delta t} + R^n \right).$$

HEAT97 uses MILU (Modified Incomplete-LU decomposition) to solve this equation. MILU [6] is a combination of the Conjugate Gradient method with a suitable preconditioner. This preconditioner is an important advantage above other solving methods.

The matrix  $A$  is decomposed in a lower and an upper triangular matrix  $L$  and  $U$  respectively. Both are having the same structure as the lower and upper parts of  $A$ . The product of  $L$  and  $U$  has almost the same structure as  $A$ , except for two extra diagonals. The elements of these diagonals are called 'fill-in elements'. If these elements are ignored, then it is possible to find  $L$  and  $U$  whose product equals  $A$ . Those ignored fill-in elements can result in unreliable solutions when those elements are rather big. To compensate this effect the fill-in is subtracted from the diagonal entries.

With this preconditioner (called  $K$ ) the algorithm used for the conjugate gradient method becomes:

1. Take  $\mathbf{x}^{(0)} = \mathbf{p}$ , and calculate  $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$  and  $\mathbf{z}^{(0)} = K^{-1}\mathbf{r}^{(0)}$

Compute for  $n = 0, 1, 2, \dots$  the vectors  $\mathbf{x}^{(n+1)}$ ,  $\mathbf{r}^{(n+1)}$  and  $\mathbf{z}^{(n+1)}$  from

2.  $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha_n \mathbf{z}^{(n)}$  with  $\alpha_n = (\mathbf{r}^{(n)}, K^{-1}\mathbf{r}^{(n)}) / (\mathbf{z}^{(n)}, A\mathbf{z}^{(n)})$
3.  $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \alpha_n A\mathbf{z}^{(n)}$
4.  $\mathbf{z}^{(n+1)} = K^{-1}\mathbf{r}^{(n+1)} + \beta_n \mathbf{z}^{(n)}$  with  $\beta_n = (\mathbf{r}^{(n+1)}, K^{-1}\mathbf{r}^{(n+1)}) / (\mathbf{r}^{(n)}, K^{-1}\mathbf{r}^{(n)})$

This algorithm has been inserted in MILU and gives us  $p^{n+1}$ . The new velocity is calculated in SOLVEP using  $u_{i,j}^{n+1} = (u_{i,j}^n + \delta t R_{i,j}^n) - \delta t \text{grad } p_{i,j}^{n+1}$ , where the gradient is discretized as

$$\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\frac{1}{2}\Delta x_i + \frac{1}{2}\Delta x_{i+1}}$$

### 3.1.5 Boundary, in- and outflow conditions

As noted earlier, the boundary conditions for the Navier-Stokes equations usually are  $u = v = 0$  on solid walls (no-slip condition). Since the velocities are not always defined on the walls because of the placement of the variables, we have to interpolate and make use of virtual mirror points. For example, consider the lower boundary ( $j = 1$ ). The horizontal velocity  $u$  is not defined *on* the wall, but half a grid point higher. So what we do is, we set  $u(i, 0) = -u(i, 1)$  so that after interpolation of these velocities the resulting velocity *on* the wall is zero.

The cells with label '8' form the inflow openings. Their velocity is set at the prescribed value (depending on the orientation of the outer wall in which the opening lies). The inflow cells are further treated the same as boundary cells.

The cells with label '7' make up the outflow opening. Such openings are currently restricted to the outer walls. The boundary conditions at such a region of the wall are prescribed by natural boundary conditions. In an outflow cell, we set the pressure at its ambient value:  $p = 0$  (thus ignoring any viscous contribution to the normal stress of the natural boundary condition).

## 3.2 Turbulent extension of HEAT97

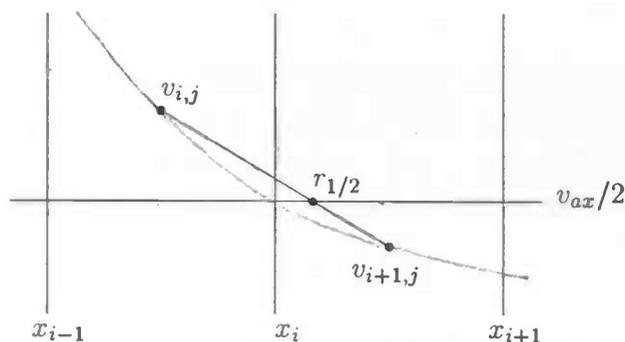
To solve the momentum equations when the jet becomes turbulent, HEAT97 needs some changes and also an extra subroutine.

The viscosity has to be placed in the center of a cell like the pressure. Then the equations

in TILDE involving terms with  $\nu$  have to be changed. Because the kinematic viscosity has become place dependent, it needs to be discretized in radial direction as follows:

$$\frac{\partial}{\partial x} \left( \nu \frac{\partial u}{\partial x} \right) = \frac{\nu \frac{\partial u}{\partial x} \Big|_r - \nu \frac{\partial u}{\partial x} \Big|_l}{\delta x}$$

The turbulence model (Prandtl's mixing-length model) is dependent of the 'half-width radius' (i.e. the radius at which the velocity has one half its axial velocity). Therefore an extra subroutine HALFW<sup>3</sup> had to be created. In this subroutine for every row  $j$  the halftime radius is determined by comparing the axial velocities of the cells to half times its value on the axis. When the value of the  $(i + 1)$ -th cell is just less and the  $i$ -th cell is just more then half the value on the axis, then by linear interpolation the half-width radius is determined, and for every cell the turbulent viscosity can be calculated.



<sup>3</sup>For a description of HALFW, see appendix A 1.4

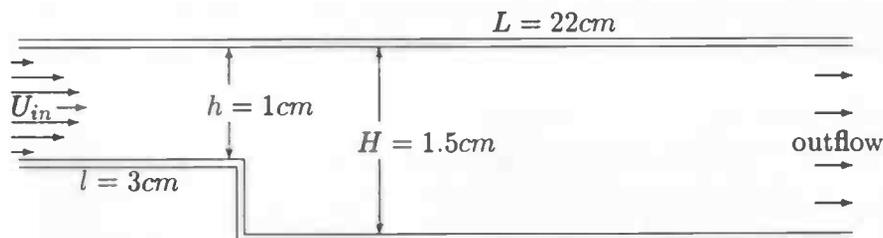
## Chapter 4

# Results

All the features that have been investigated are written down in this chapter. It gives a total review of what is done to come finally to the purpose of this master's thesis: the turbulent jet.

### 4.1 Testing HEAT97

To make sure that the program is working well, it needs a little testing. A reliable test-case is the **Backward Facing Step**. This problem was chosen by the organizers of a workshop [5], because it's suitable for every known method solving the Navier-Stokes equations, and its size is small. The geometry of the test-case is described by:



The conditions used to complete the problem are: on the walls the boundary conditions are no-slip; the Reynolds number is defined by:

$$Re \equiv \frac{U_{max} (H - h)}{\nu},$$

with  $U_{max}$  being the maximum value of the velocity profile on the inflow-side; and the inflow profile should be parabolic.

With this geometry two cases have been investigated. In both cases the kinematic viscosity  $\nu$  has been chosen as 0.01. To create the right Reynolds number, the velocity at the entrance has been varied. The grid that has been used for both cases was  $102 \times 32$  (all rectangular cells with equal distances).

The first case considered is the one with  $Re = 50$ . This means that the  $U_{in}$  must be 1.0. The resulting flow pattern right behind the step can be seen in figure 4.1.

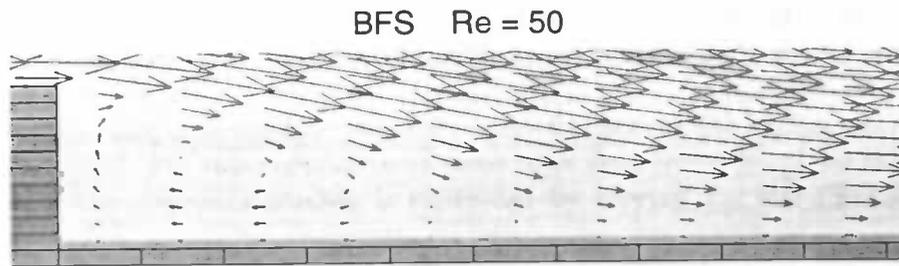


Figure 4.1:

For the second case  $Re = 150$  has been taken. Therefore  $U_{in} = 3.0$  and the resulting pattern is written in figure 4.2.

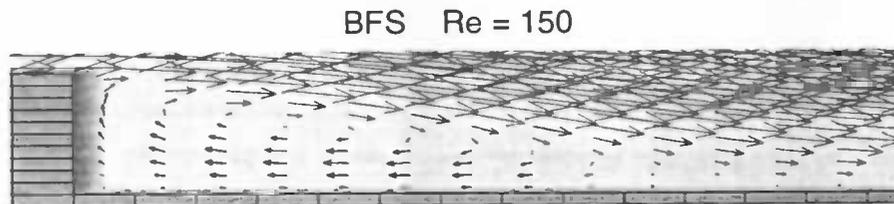


Figure 4.2:

To evaluate these runs the most important thing to view is the length of the recirculation area. To compare the results, some other results of the workshop are written down in table 4.1.

Table 4.1: Re-attachment values of other tests

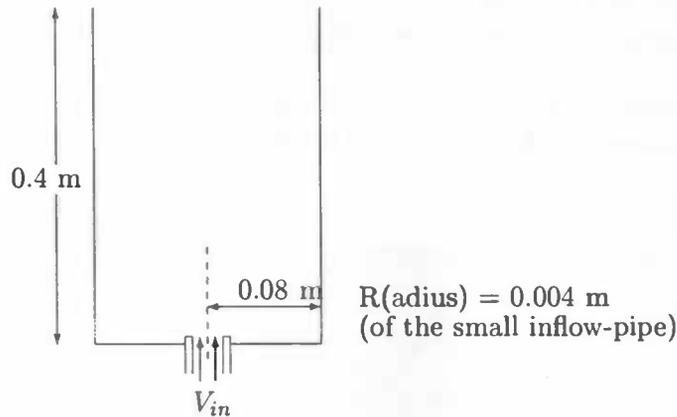
$x = R/(H - h)$	$Re = 50$	$Re = 150$
Durst and Pereira	2.54	5.24
Braza, Haminh and Cazalbou	2.74	5.05
Becker and Wachter	2.8	5.16
HEAT97	2.75	5.25

( $R$  is the position of the re-attachment point.)

The values are very much alike, so now the real problem can be considered.

## 4.2 The laminar jet

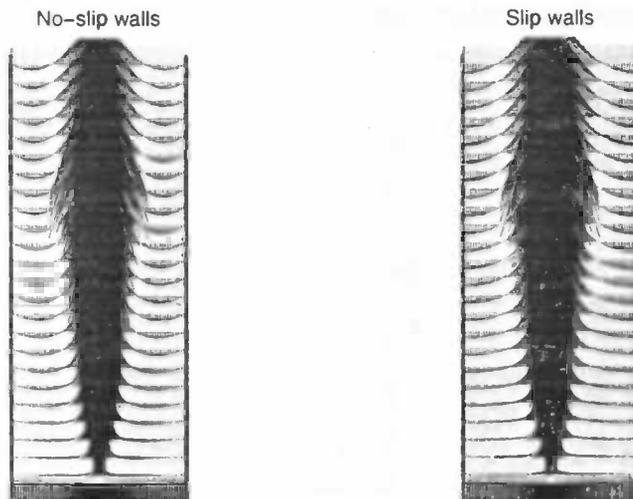
The (axisymmetrical) geometry being used for the simulation of the jet is due to Wittmer[4]. Below a sketch of the geometry is given.



The 'walls' of the flow region are discretized as outflow cells. So the flow region is regarded as an open space. Wittmer uses a big pipe, with a radius of  $0.6 \text{ m}$ , around the inflow pipe due to measurement technical reasons. The Reynolds number is defined by :

$$Re = \frac{RV_{in}}{\nu}$$

with  $V_{in}$  being the maximum value of the almost rectangular distributed inflow profile. To check if the problem is going the right way (spreading from the injection plane in a conical way), first it's done with no-slip walls. After that the walls are defined as slip walls.



The figures above are satisfying the expectations. The jet expands in a conical way from the injection point, and at the walls the prescribed conditions are as they were set. Knowing this, the real problem to be considered has to be run.

For this situation the boundary walls are all outflow openings, but at the nozzle it's necessary

to put one no-slip wall between the in- and the outflow cells. The reason for this is that if an in- and outflow cell were placed next to each other, then there would be problems due to their discretization. However the extra cell isn't strange compared to reality, because there always exists a wall of the inflow pipe at the injection plane.

According to the analytical solution there are some features of the laminar jet that can be checked by HEAT97. For this purpose three cases have been investigated, namely:  $Re = 50$ , 100 and 200. The Reynolds number is controlled by varying  $V_{in}$  and the constant  $\nu = 1.52 \cdot 10^{-3}$ .

The flow patterns are shown in figure 4.3. We can see that the jet is getting less wide when the Reynolds number increases. This is also what the mathematics of equation 2.14 predicted.

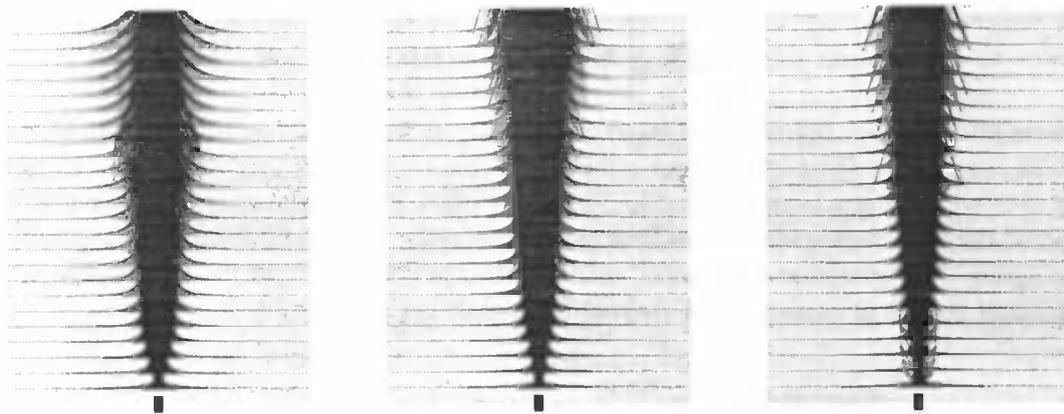


Figure 4.3:  $Re = 50, 100$  and  $200$

The data make it possible that features as axial velocity, pressure and jet angle can be plotted. In figure 4.4 the axial velocity is regarded.

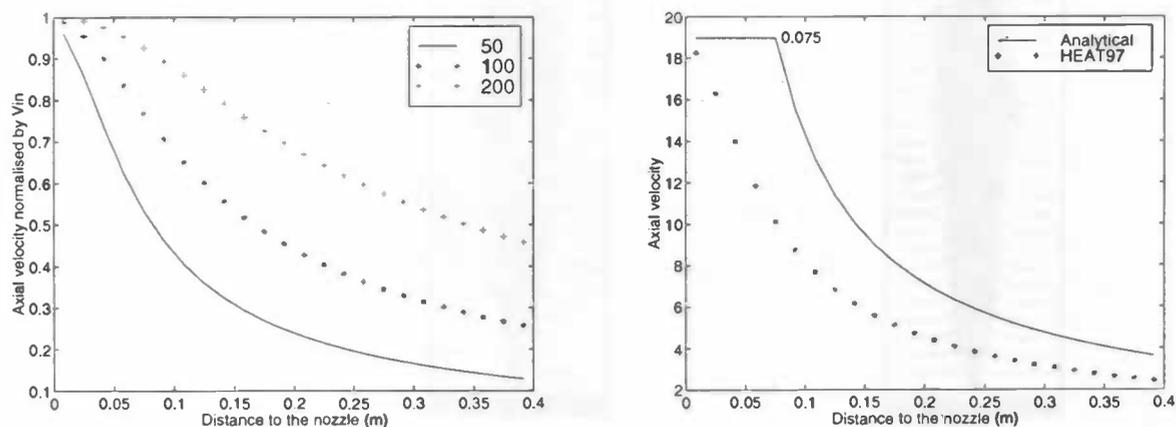


Figure 4.4: Axial velocity

To the left the axial velocity (by  $V_{in}$  normalized) has been compared for  $Re = 50, 100$  and  $200$ . For higher Reynolds numbers the axial velocity stays higher along the axis, which can be

explained by a lower spreading rate. To the right the analytical solution is compared to the results of HEAT97 for  $Re = 50$ . The analytical solution is only valid for  $z > 0.075$  (see section 2.2.2). Therefore the axial velocity in this area is defined as  $V_{in}$ . The difference between both figures is caused by the assumptions made to the analytic solution. Hereto some boundary layer theory is involved. The shape of both figures is almost similar, so the remark that the axial velocity varies reciprocally with longitudinal distance  $z$  can be made. Because the data gave a result that was remarkable it will be mentioned. The peak in the pressure in figure 4.5 is the remarkable thing.

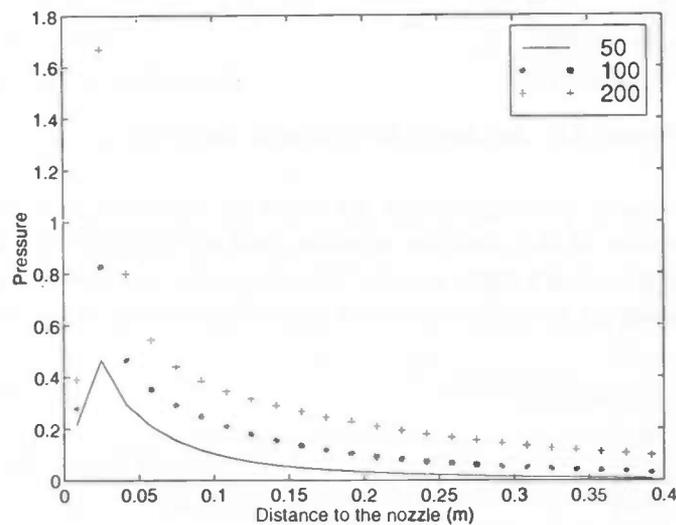


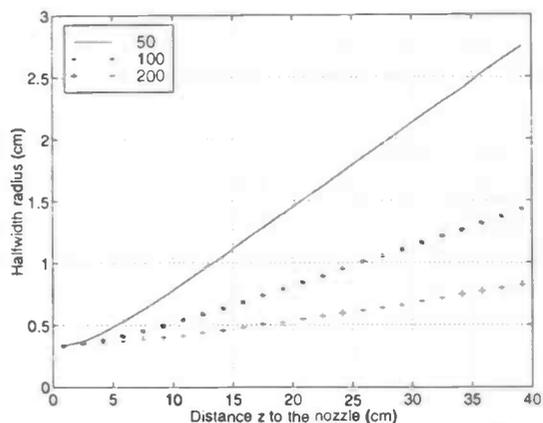
Figure 4.5: Pressure on the symmetry axis

With one extra cell inside the 'pipe', the pressure doesn't 'peak'. Therefore the reason for the peak is probably caused by the enormous pressure difference at the nozzle. With one extra cell the jet doesn't have to conquer this difference. However the side effects to this extra cell are also important. With this geometry the Reynolds number and the inflow profile are less easy to control. Because these features are more important to the real problem the alternative geometry is disregarded.

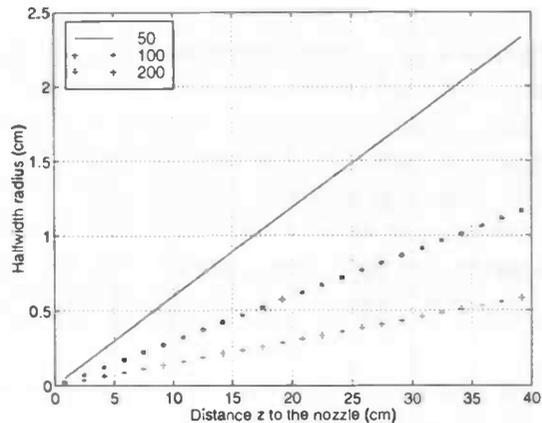
The jet angle is, as to say, a characteristic feature of the laminar jet. In figure 4.6 (see next page) the results performed by HEAT97 are compared to the values for the jet angle for the analytical solution. (The jet angle is equal to the slope of the figure.)

When only the results of HEAT97 are regarded, then the property for the jet angle, being inversely proportional to the Reynolds number, is clearly shown. The upper line is having a slope of 0.0684, and for  $Re = 100$  this value is 0.031. So the jet angle for  $Re = 100$  is approximately two times smaller as for  $Re = 50$ .

Compared to the analytic solution the values of the half-width radius are approximately 0.3cm higher (at every distance to the nozzle). This is caused by the radius of the inflow pipe, which is 0.4cm. The analytic solution doesn't take this into consideration.



Performed by HEAT97



According to the analytic solution

Figure 4.6: Jet angle for different Reynolds numbers

Not only the axial velocity profiles for the jet must be regarded, but also its cross section. For  $Re = 50$  the profiles of the analytic solution and of HEAT97 are shown in figure 4.7. The shape of both profiles don't differ much. Therefore the conclusion can be drawn that the spreading of the laminar jet in radial direction is as expected by the analytical solution.

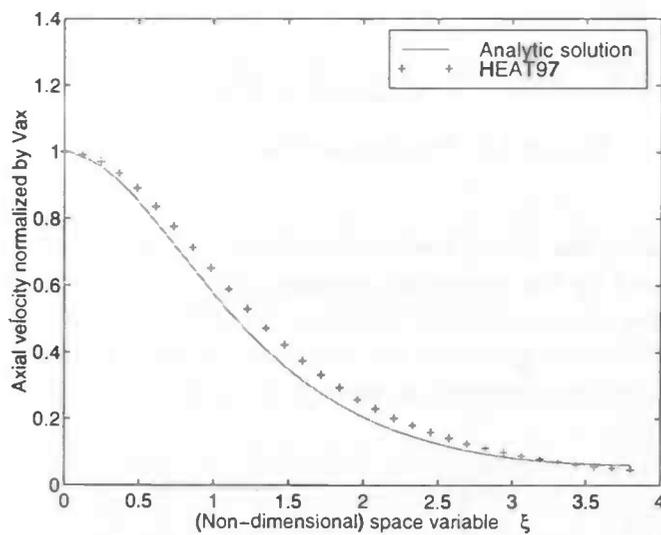
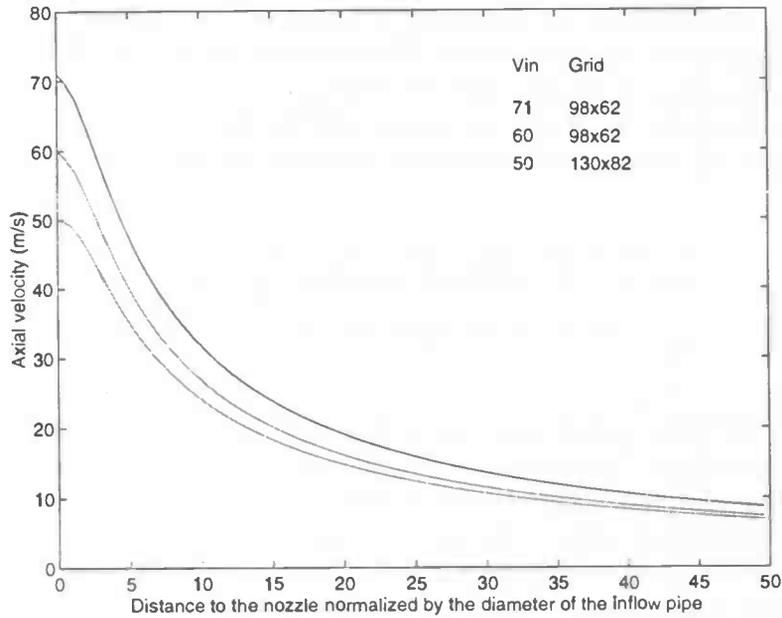


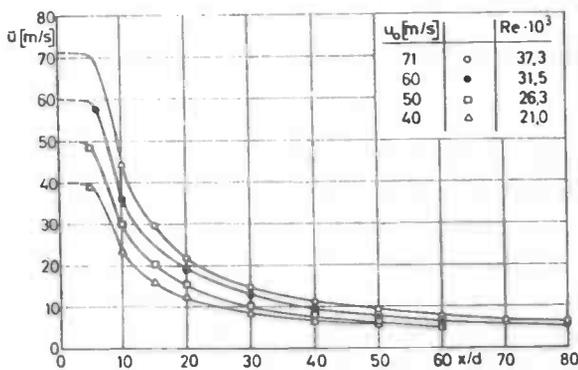
Figure 4.7: Radial profile of  $V_{ax}$  at  $z = 31.25$  cm

### 4.3 The turbulent jet

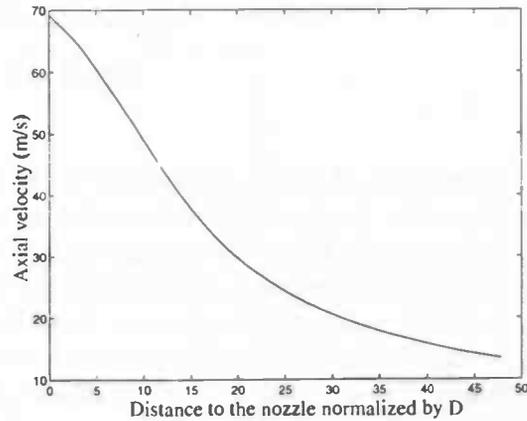
In the turbulent case the results of HEAT97 are compared with the analytic solution from Spalding (see section 2.3.2) and the measurements from Wittmer [4]. Because the jet is turbulent the effective Reynolds number is enormously increased. With a kinematic viscosity of  $\nu = 1.52 \cdot 10^{-5}$  at speeds varying from 50 to 71 m/s a few cases are investigated.



Performed by HEAT97



Measured by Wittmer



For a Low Reynolds  $k-\epsilon$  model

Figure 4.8: Axial velocity on the symmetry axis

In figure 4.8 the performance of HEAT97 is compared to Wittmer and also to a numerical result (for  $V_{in} = 71$ ) of Roel Luppés. There are clear differences between the plots. Comparing the two numerical results, it's clear that the axial velocity of HEAT97 is smaller along the

whole length of the longitudinal axis. Probably the reason for this is, that the jet spreads to fast with HEAT97, because of the effective kinematic viscosity. This gets too large due to the inaccurate computation of  $r_{\frac{1}{2}}$ .

The biggest difference with the measurements of Wittmer is that the axial velocity doesn't remain constant for a while. (Spalding also 'predicted' this constant area with his mixing-length model (see section 2.3.2)). The main reason for this difference is also the too large value for the effective viscosity at the nozzle.

Strangely enough the shape of the figures of Wittmer and of HEAT97 are very much the same. By a translation of (approximately) 4.5 along the horizontal axis the figures almost coincide. So the spreading of the jet might be okay.

The differences are not the result of a too coarse grid. In figure 4.9 the results of refining the grid is showed. For the coarser grid three cells are taken as inflow cells and five cells for the other.

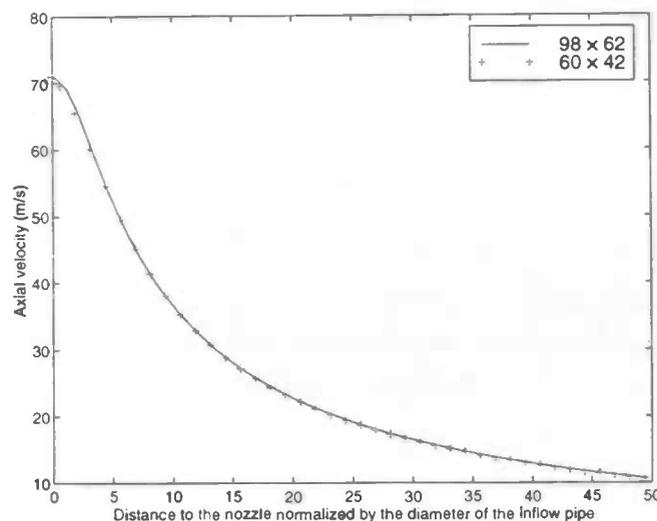


Figure 4.9: Grid refinement

The spreading of the jet is described by its jet angle. In figure 4.10 (see next page) it's shown for  $V_{in} = 71$ . The constant  $\alpha$  in the upper left corner is defined as the slope of the figures. This value is for Prandtl's mixing-length model much lower as for the analytic solution. The value deduced from the measurements is 0.088, which is close to the analytic value. So at first sight a strange thing occurs. Over the whole longitudinal length the axial velocity is lower then the measured values and the spreading of the jet is less then for Wittmer's values. Probably this is caused by the turbulence modeling at the nozzle. The value for the halfwidth radius is very inaccurate in that area, and therefore the velocity at the entrance plane isn't right.

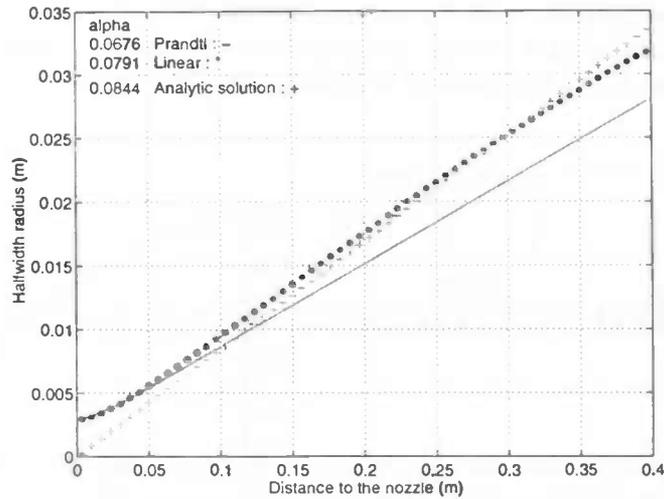


Figure 4.10: Jet angle for  $V_{in} = 71$

Also the mixing-length model used by Spalding has been tested with HEAT97. The value for alpha with this model approaches the analytical value better. However the axial velocity right behind the nozzle doesn't remain constant for a while. Again this seems to be caused by the computation of the halfwidth radius. At the inflow-pipe the velocity profile is nearly rectangular, so the velocity in e.g.  $(i, j) = (5, 2)$  is almost equal to the velocity at the symmetry axis. For the halfwidth radius the two cells which are just less and just more than half the velocity at the axis are searched. This means for row 2 that the radius of the inflow-pipe determines the halfwidth radius. When the radius of the inflow pipe is subtracted from the computed halfwidth radius the velocity does remain constant for a while. (See figure 4.11)

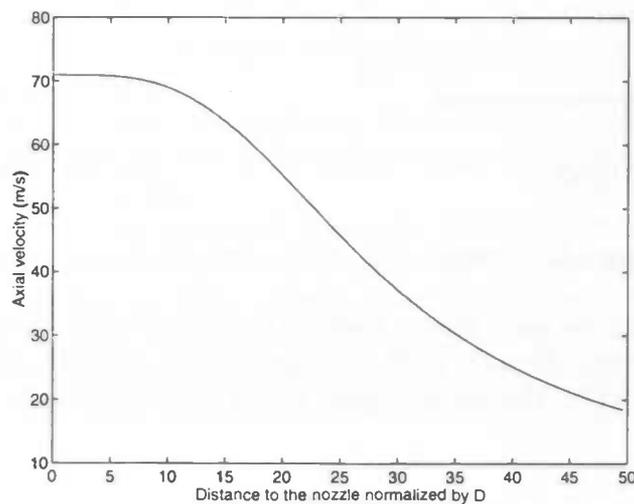
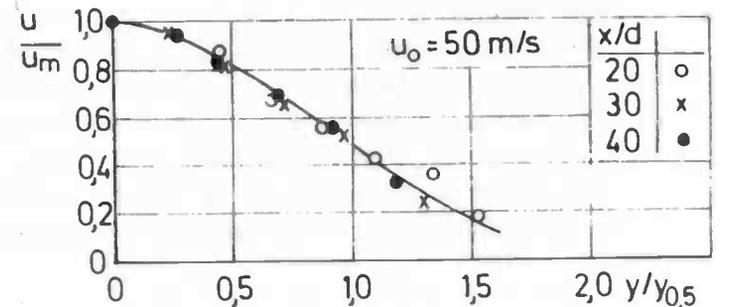
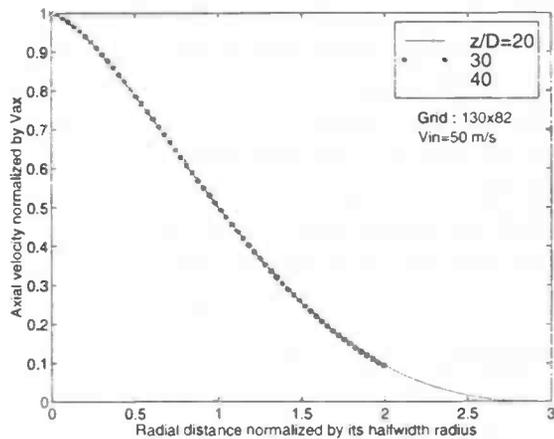
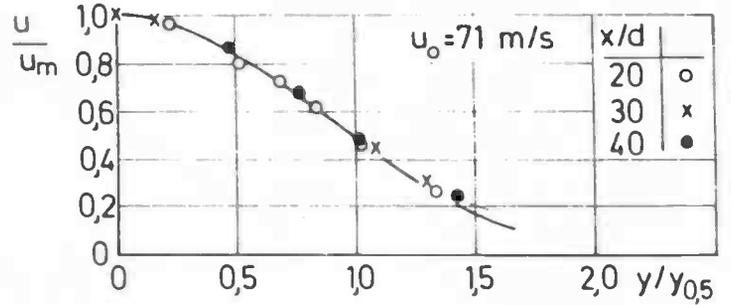
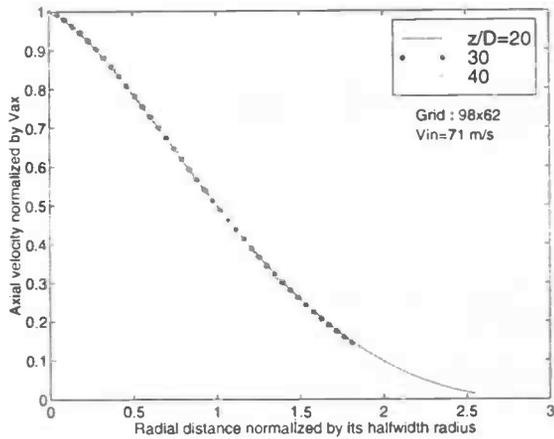


Figure 4.11: Axial velocity on the symmetry axis

This doesn't mean that such a simple model can describe the turbulent jet completely. When

we look at the axial velocity at a longer distance we see that it doesn't coincide with the measurements of Wittmer (figure 4.8). And so the subtraction is apparently just a trick to imitate the measured data near the nozzle exit.

The feature that does coincide with the measurements of Wittmer is the cross-profile of the (normalized) axial velocity at a certain distance to the nozzle. In figure 4.12 the figures are compared for different  $V_{in}$  and measured at fixed points.



(a) HEAT97

(b) Wittmer

Figure 4.12: Cross-profile of the normalized axial velocity

Both figures are having the same shape. Perhaps this is caused by the normalization in both axial and radial direction. Because both the axial velocity and the jet angle are smaller for the simulation by HEAT97, the normalization makes those differences disappear!

## Chapter 5

# Conclusions and recommendations

The purpose of this report was to try to get a more accurate and faster simulation of the turbulent jet. With respect to the time necessary for computation, we should say that HEAT97 is a time-accurate explicit method, hence computational effort is mainly dependent of the time needed by the physics. (The higher the entrance velocity, the sooner the solution becomes stationary.) Non-time-accurate (implicit) methods can be an order of magnitude faster, and shouldn't be compared directly to the present approach. Therefore no comparison of time has been included to the chapter 'Results'.

The accuracy of the solution obtained with HEAT97 is somewhat poor compared to the measurements of Wittmer. The most important reason for this is the modeling in HEAT97 of the turbulence. With Prandtl's mixing-length model it seems that the solution is only reliable for large values of  $z$ . Maybe with further changes in turbulence modeling near the nozzle it's possible to create more accurate results. For example it might be possible to define in this area a model like the one of Spalding (where the radius of the inflow-pipe was subtracted from  $r_{\frac{1}{2}}$ ). Another improvement could be obtained by looking more closely to this area. During my investigations I only considered the whole spreading of the jet. When a small area near the nozzle (say the area predicted by Spalding where the solutions don't agree to the boundary conditions at the entrance plane) is investigated, there might be a better view at the turbulence behind the nozzle.

With HEAT97 it's possible to use a finer grid near the nozzle, but at high Reynolds numbers, as used here for the turbulent jet, the CFL-number (limit of the explicit time integration) makes solving more time consuming.

At the end I would like to make a remark about the literature. According to Spalding it looks like the turbulent jet has already been analyzed and that no further investigations are necessary. To me the contrary has been 'proved', investigating the turbulent jet myself with a reliable program, as HEAT97 seemed to be. If Spalding would be right the turbulent jet could be solved by an almost linear turbulence model! This was found not to be true, so further investigations have to be made to understand the complexity of turbulent jet flows.

## Appendix A

# Program description

### A.1 FORTRAN77

#### A.1.1 Calling sequence

The calling sequence within HEAT97 can be represented as follows:

initialization	SETPAR	GRID
	SETFLD	MKGEOM
		LDSTAT
time step	INIT	BC
	TILDE	BDYFRC
		HALFW
	SOLVEP	COEFF
		MILU
		BC
	CFLCHK	DTADJ
	SVSTAT	

The post-processing routines are omitted for presentational reasons.

#### A.1.2 Common block variables

In FORTRAN, common blocks are used to make variables global. That is, the variables in a common block can be used in every subroutine in which this common block is declared, enabling subroutines to exchange data. Below, we state the variables of importance in alphabetical order.

`/COEFP/` contains the coefficients for the pressure in the Poisson equation:

<code>CC(I, J)</code> ,	: coefficients of $p_{i,j}$ ,
<code>CN(I, J)</code> ,	$p_{i,j+1}$ ,
<code>CS(I, J)</code> ,	$p_{i,j-1}$ ,
<code>CE(I, J)</code> ,	$p_{i+1,j}$ ,
<code>CW(I, J)</code>	and $p_{i-1,j}$ respectively
<code>DIV(I, J)</code>	: right-hand side of the Poisson equation

/FLUID/

Nu : laminar viscosity  
NUEFF : effective viscosity

/GRIDAR/ contains parameters involving grid-size:

X(I) :  $x$ -coordinates of the grid points  
XI(I) :  $x$ -coordinates of cell centers ( $\frac{1}{2}(x_{i-1} + x_i)$ )  
DelX(I) : distance in  $x$ -direction between two subsequent grid points  
( $\Delta x_i = x_i - x_{i-1}$ )  
Y(J) :  $y$ -coordinates of the grid points  
YJ(J) :  $y$ -coordinates of cell centers ( $\frac{1}{2}(y_{j-1} + y_j)$ )  
DeLY(J) : distance in  $y$ -direction between two subsequent grid points  
( $\Delta y_j = y_j - y_{j-1}$ )  
RX(I), RXI(I), : inverse of X(I), XI(I) and DelX(I) respectively i.e.  $\frac{1}{x_i}$ ,  
RDX(I)  $\frac{2}{x_{i-1}+x_i}$  and  $\frac{1}{\Delta x_i}$   
Circum(I) : the circumference of a circle with radius  $x_i$  (in the axisymmetric case, 1 otherwise)  
RDY(J) : inverse of DELY(J) ( $\frac{2}{y_{j-1}+y_j}$ )  
CX, CY : stretch parameters in  $x$ - and  $y$ -direction respectively  
CYL, ICYL : floating-point and integer version of the 2D/axisymmetric switch (CYL=0 for 2D and CYL=1 for axisymmetric geometries)  
IMaxUs, : number of grid points in  $x$ - and  
JMaxUs,  $y$ -direction  
IM1Us,IM2Us : IMaxUs=IM1Us+1=IM2Us+2  
JM1Us, JM2Us : JMaxUs=JM1Us+1=JM2Us+2

/ORGA/ is used for cell labeling:

NF(I,J), : contains cell labels for the current time level

/ORGA2/ contains the parameters concerning the iterative proces

IMilu : numerical model parameter  
Iter : iteration number of the conjugate gradient method  
ItMax : maximum iteration number  
ItSum : iteration number of the whole proces  
Epsi : Poisson convergence criterion  
Alpha : upwind parameter

/PHYS/ contains pressure and velocities at the current and the previous time level. An N means 'at the previous time level':

F(I,J) : VOF (Volume Of Fluid)/ indicator function  
U(I,J), UN(I,J): horizontal/radial velocity  
V(I,J), VN(I,J): vertical/axial velocity  
W(I,J), WN(I,J): azimuthal velocity  
P(I,J), PN(I,J): pressure  
VMAX : maximum attained velocity

PMIN, PMAX : minimum and maximum pressure  
(The latter three variables are used for scaling during post-processing.)

/TIMES/ contains parameters related to time levels and steps:

Cycle : time step number  
T : current time  
Delt : time step  
DeltMx : maximum allowed time step  
TFin : end time  
TStart : start time

### A.1.3 Subroutines

Below, a short description of HEAT97's subroutines is given.

BC : sets boundary conditions for velocity components  
BDYFRC : computes the apparent body force  
HALFW : computes the (axisymmetric) halftime radius and adjusts it to the place-varying kinematic viscosity  
CFLCHK : monitors CFL number and sets flag for time-step adjustment  
COEFF : defines coefficient matrix for Poisson equation including boundary conditions at wall and obstacles  
DTADJ : halves or doubles time-step (old time-step will be repeated)  
GRID : makes (non-uniform) grid  
INIT : starts new time-step  
MILU : solves Poisson equation  
PRT : prints and writes results  
SETFLD : initializes fluid configuration  
SETPAR : reads input file  
SOLVEP : organizes pressure calculation and updates velocity  
TILDE : integrates momentum equations  
AVS : generates output to be processed by AVS  
GNUPLT : writes data to a pipe to be displayed 'live' by GNUplot  
PRTFLD : produces a sequence of characters representing the entire geometry including obstacles (#=obstacle, \*=fluid, I=inflow and 0=outflow)  
PRTCNF : creates datafile for MATLAB use  
MKGEOM : reads geometry file  
SVSTAT : writes restart file  
LDSTAT : reads the restart file (produced by SVSTAT)  
MKSTRL : keeps track of specified particles by integrating velocities  
FLXOUT : determines and saves the fluxes through specified areas  
FRCOUT : integrates pressure on the walls in both  $x$ - and  $y$ -direction and writes the resulting force to file  
MPTOUT : interpolates velocities in points specified in the geometry file and saves them

#### A.1.4 HALFW

To create the turbulent extension of HEAT97, the following subroutine was used to calculate the viscosity in a cell center.

```
SUBROUTINE HALFW
IMPLICIT NONE

INTEGER IMAX,JMAX
PARAMETER (IMAX=130, JMAX=130)

COMMON /GRIDAR/ X,XI,DelX, Y,YJ,DeLY, RX,RXI,RDX,Circum, RDY,
1  CX,CY, CYL,ICYL, IMaxUs,JMaxUs,IM1Us,JM1Us,IM2Us,JM2Us
REAL X(IMAX),XI(IMAX),DelX(IMAX), Y(JMAX),YJ(JMAX),DeLY(JMAX),
1  RX(IMAX),RXI(IMAX),RDX(IMAX),Circum(IMAX),RDY(JMAX),CX,CY,CYL
INTEGER ICYL, IMaxUs,JMaxUs,IM1Us,JM1Us,IM2Us,JM2Us

COMMON /PHYS/ F,U,UN,V,VN,W,WN,P,PN,VMAX,PMIN,PMAX
REAL F(IMAX,JMAX),U(O:IMAX,JMAX),UN(O:IMAX,JMAX),
1  V(IMAX,O:JMAX),VN(IMAX,O:JMAX),W(IMAX,JMAX),WN(IMAX,JMAX),
2  P(IMAX,JMAX),PN(IMAX,JMAX),VMAX,PMIN,PMAX

COMMON /FLUID/ Nu,NUEFF
REAL Nu,NUEFF(IMAX,JMAX)

COMMON /ORGA/ NF
INTEGER NF(IMAX,JMAX)

REAL rhw,dvdxT,dvdxB,dvdx,dvdxq,dvdxp
INTEGER i,j

DO J=2,JM1Us
DO I=2,IM1Us
  IF (.5*VN(2,J).LT.VN(I,J) .AND. .5*VN(2,J).GT.VN(I+1,J)) THEN
    RHW=XI(I+1)-(XI(I+1)-XI(I))*
*      (.5*VN(2,J)-VN(I+1,J))/(VN(I,J)-VN(I+1,J))
    ENDIF
  ENDDO
DO I=2,IM1Us
  DVDXT=(VN(I+1,J)-VN(I,J))/(XI(I+1)-XI(I))
  DVDXB=(VN(I+1,J-1)-VN(I,J-1))/(XI(I+1)-XI(I))
  DVDX=0.5*(DVDXT+DVDXB)
  NUEFF(I,J)=Nu+0.03515625*RHW*RHW*ABS(DVDX)
  ENDDO
  ENDDO
DO I=2,IM1Us
  NUEFF(I,1)=NUEFF(I,2)
```

```

        NUEFF(I, JMaxUs)=NUEFF(I, JM1Us)
ENDDO
DO J=2, JM1Us
    NUEFF(1, J)=NUEFF(2, J)
    NUEFF(IMaxUs, J)=NUEFF(IM1Us, J)
ENDDO
NUEFF(1, 1)=NUEFF(2, 2)
NUEFF(1, JMaxUs)=NUEFF(2, JM1Us)
NUEFF(IMaxUs, 1)=NUEFF(IM1Us, 2)
NUEFF(IMaxUs, JMaxUs)=NUEFF(IM1Us, JM1Us)

RETURN
END

```

## A.2 Input files

### A.2.1 Main input

HEAT97 uses at least one input file, the main input file. To create a special geometry, another input file can be used. This one is described in the next section. The main input file is called `heat97.in` and has the following structure:

```

***** tank geometry *****
icyl      Xmin      Xmax      Ymin      Ymax      spec.geom.
  1        0.0      0.08      0.0      0.4      1

***** grid definition *****
iMaxUs    jMaxUs    cx      cy      xpos      ypos
  98      62      0.0    0.0    0.0      0.0

***** liquid properties *****
Nu      IHeat  Pran  Rayl(-g*beta)
1.52e-5  0      0.71  0.034

**** gordijnparameters *****
Vgordyn  Tgordyn  Tbinnen  Tbuiten  Tvloer (<0 = adiab.)
  4.0    30.0   20.0    5.0    -15.0

***** body forces and external motion: 2D *****
Gx  TxOn  TxOff  u0      Gy  TyOn  TyOff  v0
0.0 0.0  100  0.0    0.0 0.0  100.0  0.0
Ampl  Freq  Angle
0.0  0.0  0.0
Rpm  DelOme  TwUp  TwDown  x0  y0
0.0  0.0    0.0  10.0   0.0  0.0

```

```

**** body forces and external motion: axisymmetric ****
Gy   TyOn  TyOff  v0     Ampl  Freq
0.0  0.0   100   0.0    0.0   0.0
Rpm  DelOme TwUp   TwDown
0.0  0.0   0.0   100

**** boundary conditions and inflow characteristics ****
left  right  top  bottom  Amplin  Freqin  IPIn  PIn
1     7     7   7       71.0   0.0     0    -1.0

**** upwind parameter and Poisson iteration parameters ****
Alpha  Epsi   ItMax
1.0    1.0e-4  100

**** time step and restart control ****
TFin  DelT   PrtDt/DelT  svst  svdt
10    7.0e-8  1000        1     1.0

**** print/plot control ****
gnu  matlab  avs  vofmat  velo  forces
0    0       0    1       0     0

**** stream lines ****
nrx  nry  nrdt   xps  yps  xqs  yqs  t  dt/delt
3    3    0     4.0 -1.0  5.0  1.0  0.1  1

**** fluxes ****
number of fluxes to be printed (flux01.out...flux#.out are created)
5
p1    p2    p3    hor
0.0   0.08  0.4   1
0.0   0.08  0.007 1
0.0   0.4   0.08  0
0.004 0.08  0.007 1
0.0   0.004 0.007 1

```

The input file will be explained following the above example. The jet-stream problem is not temperature dependent, and there's no buoyancy. Because of those features some parts aren't important and must be set to zero.

In the *tank geometry* section the following parameters have to be set. *icyl* is the switch between 2D and axisymmetric geometries (0 for two dimensional, and 1 for axisymmetric calculations). (*Xmin*, *Ymin*) and (*Xmax*, *Ymax*) are the lower left and the upper right corner of the tank respectively. (The same units must be used for all parameters, e.g. if the values here are in meters, then the velocities should be given in meters per second. The units don't necessarily have to be SI units, but have to be consistent with each other!) To make more

complex geometries `spec.geom.` should be set to 1. Such a complex geometry has to be defined in the file `heat97.geo` (see A.2.2).

The grid is defined by the parameters in *grid definition*. `iMaxUs` and `jMaxUs` are the numbers in  $x$ - and  $y$ -direction respectively. When `cx` or/and `cy` are greater as 0 stretching is used from position (`xpos`, `ypos`), where the cells are getting bigger when they get further away from the initialized point.

The *liquid properties* used in the case of the jet-stream, are only `Nu` and `IHeat`. The first is the kinematic viscosity defined as  $\mu/\rho$ , and the second is used to switch the heat transfer off. The other two properties are used when `IHeat` is equal to 1.

Because the next two sections are used to define body forces and external motion, the only thing that should be taken care of is that all the values are initialized as zero.

Specified next are the *boundary conditions and inflow characteristics*. For every side of the tank, the desired treatment can be stated here. Values 1 or 2 set the boundary condition to slip or no-slip respectively, while values 7 or 8 make the entire side an opening for out- or inflow respectively. The rate of inflow is set at `Amplin`, and can eventually be made to oscillate with frequency `Freqin`. Because the pressure is prescribed as zero for outflow openings, `IPIn` is set 0 (i.e. the pressure has not to be changed at an outflow side/opening, `PIn` would be the new value for  $p$ ).

The numerical parameters are defined next. `Alpha` is de upwind parameter, which is 0 for central and 1 for full upwind discretization. With `Epsi`, the Poisson convergence criterion can be controlled. `ItMax` is the maximum number of iterations that HEAT97 is allowed to perform in one time step.

In the section *time step and restart control* the endtime `TFin` and the initial time step `DeIT` can be specified. The latter may be reduced or doubled by HEAT97 if necessary and possible. Also embedded in this section is the control of the frequency at which the output is written to the file. `Prtdt` is the time between two consecutive printouts to the screen and between two calls to the subroutines that produce all sorts of output (see section A.3). `20*Prtdt` is the time between two consecutive large printouts to the screen. At the end of the line, parameters for the making of 'backups' are specified. `svst` can have three values: 0 if no restart backups are required, 1 to save the program state every `svdt` time units, and 2 if a saved state from a previous run is to be read at startup, and after that the execution should proceed as with `svst=1`.

The next three sections of the input file are devoted to HEAT97's output.

First there is the *print/plot control* section. It consists of six switches used to choose the kind of output HEAT97 should produce. All the switches should be either 1, to enable, or 0 to disable the output option. The following abbreviations appear:

`gnu` : live gnuplot annimation of liquid configuration  
`matlab` : velocity and pressure data in Matlab format  
`avs` : velocity and pressure data in AVS format  
`vofmat` : VOF-function, velocity and pressure/temperature (for Matlab)  
`velo` : additional velocity and pressure output in 'heat97.out'  
`forces` : pressure forces

Further one can follow the paths of a particle in a fluid. This can be done in *streamlines* where a box can be created by (xps,yps) and (xqs,yqs). The number of particles placed in that box is defined by  $nrx \times nry$ . These particles are followed, starting at  $t=t$  and after every  $dt/delt$  time step, 'new' particles are released until the number of releases has reached  $nrdt$ . The data are written to *stream.out*. If one of the first three parameters is zero, then no streamlines are created.

The last section of the input file contains a possibility to compute the flux at a certain 'cross-section'. To that first one has to define the number of fluxes that have to be determined. When the cross-section is horizontal *hor* should be 1, and has start point (p1,p3) and end point (p2,p3); for *hor*=0 with start (p3, p1) and end (p3,p2) a vertical line is created. The resulting fluxes are written to *flux##.out*.

### A.2.2 Weird geometry

The following input file *heat97.geo* was used to create Wittmer's jet. Not all the possibilities had to be used, but they will also be explained.

```
JET
Icyl  Xmin  Xmax  Ymin  Ymax
1     0.0   0.08  0.0   0.4
cirkelbogen  lijnstukken  instromen  uitstromen
0           1           1           0
cirkelbogen:
  xm   ym   r   teta1  teta2  kant links rechts boven onder meetpntn
lijnstukken:
  x1   y1   x2   y2  kant links rechts boven onder meetpntn
0.004 0.0  0.005  0.0  0   0   0   0   1   0
instromen:
  p1   p2   kant (links=1, rechts=2, boven=3, onder=4)
0.0   0.004   4
uitstromen:
  p1   p2   kant (links=1, rechts=2, boven=3, onder=4)
```

The first input line (for *Icyl*, *Xmin*, etc.) must be defined, but will be overruled by the same parameters defined in *heat97.in*. To create all kinds of geometries one can use *cirkelbogen* (circular arcs), *lijnstukken* (lines), *instromen* (inflow) or *uitstromen* (outflow). The number of those objects used, must be defined in the next line. After that all the objects can be specifically defined. When an object doesn't occur, the input line for this object should be empty (as in the example above).

First the circular arcs can be defined. They have to be described by their center (*xm*,*ym*), their radius *r*, and their start and end angle *teta1* and *teta2* respectively. Then the information about where the obstacle cells have to be put must be given: 0 results in an arc filled at the side of the center of the circle, and 1 if its complement has to be filled. The next four arguments tell HEAT97 towards which outer wall the object is to be extended (either 0 or 1). The last argument gives a possibility to create extra information about the velocity nearby

an object. Every line of the 'arcs' section represents an arc.

Secondly lines (or rectangles) can be created. They are defined by their begin (x1,y1) and end (x2,y2) (or by their opposite corners (x1,y1) and (x2,y2)). The 'side' argument can have three different values:

- 0 : the object is situated underneath the line, or - in case of a vertical line - to the left of the line,
- 1 : the object is above the (or - for vertical lines - to the right of) the line,
- 2 : creates a (filled) rectangle.

The other arguments do the same as they do for circular arcs.

Finally, the in- and outflow openings have to be stated. Those can only be created at the outer walls, so three arguments will suffice: p1 and p2 to determine the coordinates of the opening, and side is here used to determine the side at which the opening should be created (1 = left, 2 = right, 3 = upper, 4 = lower).

### A.3 Output files

Only the output files used will be described here.

heat97.out provides the user of most of the information about the different values during the simulation. It contains the coordinates of the grid points, comments about the convergence, evolution of the time-step, the number of iterations required, etc. The values of the velocity, pressure and viscosity are also written down, but to use these for post-processing the next output file is created.

uvpf####.dat contains information about the velocity, pressure and the viscosity. On every PrtDt/Delt time steps an uvpf####.dat is created. Each line is filled with five numbers, and there are as many lines as there are cells.

flux##.out is a data-file with two collums; the first one with the time and the second with the calculated flux so far.

# List of symbols

Below a list of the symbols used in this master's thesis is given with a short description. For some variables, a discrete version is given. The subscript  $i,j$  refers to the number of the cell and the superscript  $n$  denotes the point time.

$x, y, x_i, y_j$	= coordinates in horizontal and vertical direction respectively
$\Delta x_i$	= distance between two consecutive coordinates in $x$ -direction ( $\Delta x_i = x_i - x_{i-1}$ )
$z, r$	= axial and radial coordinates respectively
$u, v, u_{i,j}^n, u_{i,j}^n$	= velocity in horizontal and vertical direction respectively, or velocity in radial and axial direction respectively
$p, p_{i,j}^n$	= pressure
$\rho$	= density
$\mu$	= dynamic viscosity

# Bibliography

- [1] B. de Groot. SAVOF96 - Simulation of free-surface liquid dynamics in moving complex geometries. Master's thesis, Rijksuniversiteit Groningen, 1996.
- [2] D.B. Spalding. *Combustion and mass transfer*. Pergamon Press, 1979.
- [3] H. Schlichting. *Boundary-layer theory*. McGraw-Hill Book Company, 1987.
- [4] V. Wittmer. Geschwindigkeit und Temperatur in einer turbulenten Freistrahldiffusionsflamme. PhD thesis, Universität Karlsruhe, 1980.
- [5] K. Morgan, J. Periaux, F. Thomasset, a.o. *Analysis of laminar flow over a backward facing step*. Vieweg, 1984.
- [6] E.F.F. Botta. *Eindige-differentiemethoden*. Rijksuniversiteit Groningen, November 1992. Lecture notes
- [7] A.E.P. Veldman. *Numerieke Stromingsleer*. Rijksuniversiteit Groningen, March 1994. Lecture notes