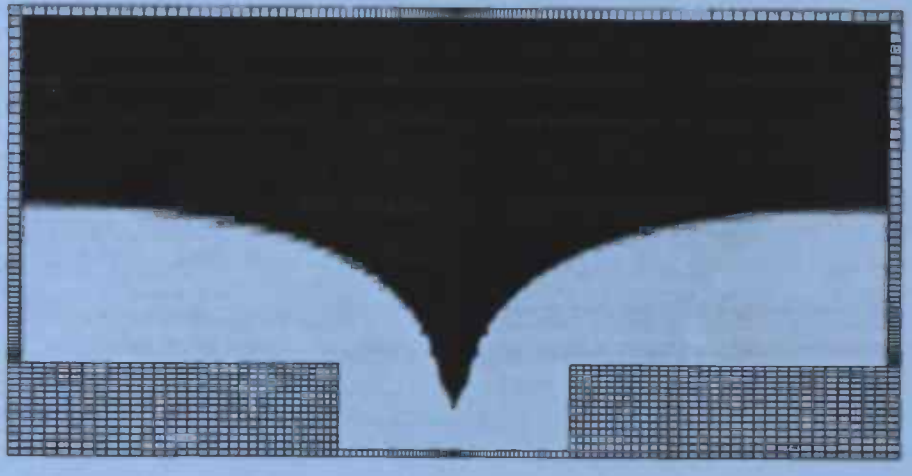
Numerical Simulation of a Two-Dimensional Dip in an Oilsump

Adri Hofste



Department of
Mathematics

RuG

Master's thesis

Numerical Simulation of a Two-Dimensional Dip in an Oilsump

Adri Hofste

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

August 1998

SAVOF96 is based on the computer program SAVOF that has been developed at the National Aerospace Laboratory NLR under contract with the Netherlands Agency for Aerospace Programmes NIVR.

Preface

This thesis is a report of my graduation work at the mathematics department of the Rijksuniversiteit Groningen. Last January, I began working on this assignment under the supervision of Ir. J.J. Nies of Wärtsilä NSD Nederland BV and Prof. dr. A.E.P. Veldman of the Rijksuniversiteit Groningen. During this period, they gave me advice and supported me very well, what finally lead to this report. To them I wish to express my sincere gratitude. Furthermore I like to thank my parents and my brother who kept faith in me during this long period of time.
In this report an attempt is made to determine the parameters that influence the height at which a dip is formed that can occur when a fluid is draining from a tank. Hereto simulations were performed using the computer program SAVOF96.
I hope that the reader has a pleasant time studying this report.

Adri Hofste
Groningen, August 1998

Contents

Chapter 1

Introduction

When a fluid is withdrawn from an open tank through a hole located in the centre of the bottom, like in a sink or a swimming-pool, sudden formation of a dip is often observed in the centre of the free surface. Most of the time, dip formation causes no problems like in a sink. Sometimes it is inconvenient because of the noise. In the oilsump of an engine, a dip can also occur.

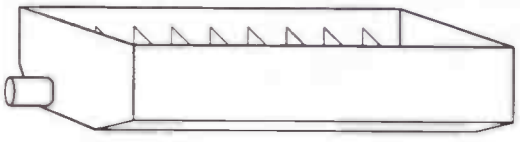


Figure 1.1: *Oilsump.*

In the Diesel engines which Wärtsilä NSD Nederland BV produces, it is also possible to use the oilsump as a lub oil-tank. In figure 1.1 we see a drawing of an oilsump. The sump is divided into different compartments by half-open bulkheads. The oil is dripping out of the engine into the sump and flows over and/or through the bulkheads to the compartment where the drain-pipe is situated. When a dip forms, air can also flow into the pipe. A mixture of air and oil will then lubricate the engine, which should be avoided.

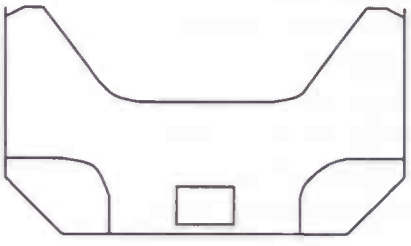


Figure 1.2: *Half-open bulkhead.*

Therefore we want to determine the height at which a dip forms. This height will be defined as the critical height. Furthermore it is important to find out which parameters influence

the critical height. Our ultimate goal is to derive a simple formula for the critical height containing all the important parameters. This formula can then be used to determine the minimum initial height at which the oil level must be set.
A means of describing fluid flow is available in the form of the *Navier-Stokes* equations. These are second order partial differential equations that can only be solved analytically for simplified cases. The two-dimensional Navier-Stokes equations are numerically solved in the program SAVOF96. This program is used to simulate the draining of a tank.
This report describes the underlying mathematics, the numerical model used to solve the equations and the results of the simulations. For a short description of SAVOF96 the reader is referred to the Appendix.

In the past, several papers were published concerning this problem or related problems. B.T. Lubin and G.S. Springer [2] studied the axisymmetric withdrawal of fluids from a circular tank and found a relation between the critical height, the volume flow rate and the density ratio. Q. Zhou and W.P. Graebel [3] also investigated the draining of fluids from an open tank under the assumption of potential flow. Their numerical results showed two different phenomena depending upon the volume flow rate and initial conditions. When the tank is rapidly drained, a dip forms at the centre of the free surface and extends into the hole very quickly, as observed by Lubin and Springer. For a slowly draining tank, a jet forms in the centre of the depression region. These papers form the basis for our studies. Whether these results are useful to us is discussed in chapters 2 and 4.

Chapter 2

Mathematical model

When confronted with a physical problem, one must first determine an appropriate mathematical model.

2.1 Oilsump

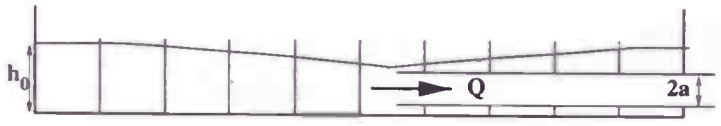


Figure 2.1: *Cross-section of a sump.*

As mentioned in the introduction, the oilsump is divided into several compartments by half-open bulkheads. Figure 2.1 shows us a cross-section of an oilsump. h_0 is the initial height of the oil, Q the volume flow rate and a is the radius of the drain-pipe. The flow in the sump can be characterised by the Froude number F, given by

$$F = \frac{Q}{a^2(g \cdot a)^{1/2}}$$

where g denotes the gravitational acceleration (9.81 m/s^2). According to Lubin and Springer the volume flow rate and the drain radius are the important parameters that influence the height at which a dip forms. Therefore this definition of the Froude number is chosen. Zhou and Graebel use $F = \frac{Q}{R^2(g \cdot R)^{1/2}}$, where R is the radius of the tank and is taken as a constant. They use the Froude number to control the volume flow rate. Hocking and Vanden-Broeck [4] introduced in their paper $F = \frac{U}{(g \cdot H)^{1/2}}$, where U is he velocity of the free surface and H is the depth of the fluid. They were not interested in the height at which a dip forms, but only in what happens just above the drain. So these Froude numbers are less useful for our model.

For the 9L38 engine built by Wärtsilä NSD we have the following values of the parameters:

volume flow rate : $Q = 0.0347\ m^3/s$
diameter drain-pipe : $2a = 0.162\ m$
distance between two bulkheads : $2R = 0.6\ m$

Using these values, the Froude number corresponding with this engine is :

$$F = 5.9$$

Our calculations will be done for Froude numbers in the neighbourhood of the above mentioned value. This means for Froude numbers in the range 4 to 9. Zhou and Graebel use Froude numbers, translated to our definition of it, larger than 28. So their results lie in a different range then ours, but we can still check if the observed phenomena also occur in our simulations. The formula of Lubin and Springer can be used in any range, what means that it might be very useful to us.

There is a difference between the two-dimensional and the three dimensional dip (whirlpool). In the three-dimensional case, rotational velocities are present whereas in the two-dimensional case they are not dealt with. The influence of these rotational velocities will be neglected.

2.2 Navier-Stokes equations

Consider a tank from which oil is draining. Fluid motion in this tank can be described by the following two conservation laws :

- conservation of mass :

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

- conservation of momentum :

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = \rho F_x + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y},$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} = \rho F_y + \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y},$$

where F_x and F_y are the components of an external force in x and y direction. $\rho(x, y)$ is the density and $\sigma = (\sigma_{ij})$ is the stress tensor. The stress for incompressible Newtonian fluids is linearly related to the rate of deformation. This is the case we study here, so we take

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

From this the 2D incompressible Navier-Stokes equations can be derived.
When the influence of external forces is neglected they look like this :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{2.1}$$

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

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

Here the kinematic viscosity $\nu = \frac{\mu}{\rho}$ was used. For later use, it is convenient to write the Navier-Stokes equations in vector notation :

$$\begin{aligned} \text{div}\boldsymbol{u} &= 0, && (2.4) \\ \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \text{grad})\boldsymbol{u} &= -\frac{1}{\rho} \text{grad p} + \nu \,\text{div grad}\boldsymbol{u}. && (2.5) \end{aligned}$$

2.3 Boundary equations

The Navier-Stokes equations have to be completed with boundary conditions. In this problem we have four types of boundaries : solid walls, free surfaces and in- and outflow boundaries.

2.3.1 Solid walls

The oil in the tank is surrounded by solid walls. The boundary conditions for solid walls usually are $u = v = 0$. Physically, these equations represent two phenomena: No oil can flow through the walls and the fact that oil sticks to the walls due to viscosity.

2.3.2 Free surface

For a free surface, the following boundary conditions are used, representing continuity of normal and tangential stresses, respectively :

$$\begin{aligned} -p + 2\mu \frac{\partial u_n}{\partial n} &= -p_0 + 2\gamma H, && (2.6) \\ \mu\left(\frac{\partial u_n}{\partial t} + \frac{\partial u_t}{\partial n}\right) &= 0, && (2.7) \end{aligned}$$

where $u_n = \boldsymbol{u} \cdot \boldsymbol{n}$ denotes the velocity in the direction normal to the surface and $u_t = \boldsymbol{u} \cdot \boldsymbol{t}$ the velocity in tangential direction. p_0 is the pressure of the air above the oil, γ is the surface tension and H represents the curvature of the surface. It is also necessary to keep track of the free surface. Therefore we use an indicator function $F(x, y)$. $F = 1$ if there is fluid present and $F = 0$ elsewhere. This indicator function satisfies :

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + (\boldsymbol{u} \cdot \text{grad})F = 0.$$

The discretized version of the indicator function is discussed in section 3.4.

2.3.3 In- and outflow boundaries

At an inflow boundary the velocity $\boldsymbol{u}$ is given : $\boldsymbol{u} = \boldsymbol{u}_{in}$. For the draining of a tank we model the required outflow area as an inflow area with a negative velocity. By doing this, we can prescribe the velocity and therefore the volume flow rate Q.

2.4 The axisymmetric case

SAVOF96 is capable of simulating fluid flow in axisymmetric containers, which we use most of the time. This is slightly different from the two dimensional case. For axisymmetric calculation an extra dimension is introduced, namely the *azimuthal direction*. The conservation laws will be written using the cylindrical co-ordinates (r, φ, z), where w denotes the azimuthal velocity:

Conservation of mass:

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

Conservation of momentum:

$$\begin{aligned}
\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + v\frac{\partial u}{\partial z} - \frac{w^2}{r} &= -\frac{1}{\rho}\frac{\partial p}{\partial r} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial u}{\partial r}) + \frac{\partial^2 u}{\partial z^2} - \frac{u}{r^2}\right), \\
\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial r} + v\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial v}{\partial r}) + \frac{\partial^2 v}{\partial z^2}\right), \\
\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial r} + v\frac{\partial w}{\partial z} + \frac{uw}{r} &= \nu\left(\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial w}{\partial r}) + \frac{\partial^2 w}{\partial z^2} - \frac{w}{r^2}\right).
\end{aligned}$$

The axisymmetric approach is quite different compared to the real situation. If we take a look at figure 1.1, we see an oilsump as it is in reality. This oilsump, or more specific one compartment, is modelled as a cylindrical tank with a hole in the centre of the bottom. Therefore the axisymmetric approach does not give an accurate description of the real situation, but is more like a rough approximation of it. It is still useful though to determine the parameters that influence the critical height and that is the purpose of this study.

Chapter 3

Numerical model

After having handled the mathematical model the numerical model used by SAVOF96 is discussed.

3.1 The Poisson equation for the pressure

In this section a Poisson equation is derived from the incompressible Navier-Stokes equations. SAVOF96 uses the Poisson equation to compute the pressure. If the density is normalised ($\rho \equiv 1$) then the equations 2.4 and 2.5 can be written as follows :

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

$\boldsymbol{R}$ contains all of the convective, diffusive and external forces :

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

If we apply the forward Euler method to the above equations we get :

$$\text{div}\boldsymbol{u}^{n+1} = 0, \tag{3.1}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\delta t} + \text{grad p}^{n+1} = \boldsymbol{R}^n. \tag{3.2}$$

Here $n+1$ and n denote the new and old time level. δt is the time step. The pressure in (3.2) must be computed in such a way that equation 3.1 is satisfied. This can be done by combining these two equations. (3.2) can be written as

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

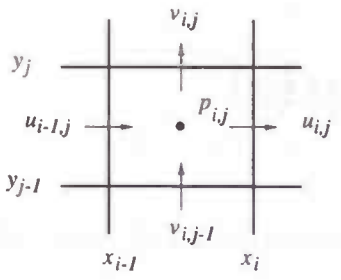
If we substitute this into (3.1), we obtain

$$\text{div grad p}^{n+1} = \text{div}(\frac{\boldsymbol{u}^n}{\delta t} + \boldsymbol{R}^n).$$

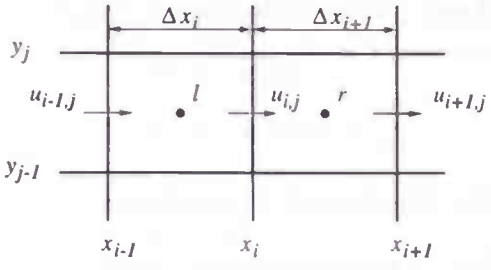
This is called the *Poisson equation* for the pressure.

3.2 Discretization

SAVOF96 uses a Cartesian grid for its discretization. The variables are placed according to the well known *Marker-And-Cell* (MAC) method. The horizontal velocity u is placed in the middle of the vertical sides of the cell, the vertical velocity v in the middle of the horizontal sides and the pressure p in the cell centres.



To solve the Poisson equation, first the momentum equation is integrated, i.e. $\frac{\boldsymbol{u}^n}{\delta t} + \boldsymbol{R}^n$ is calculated. We will now explain how the convective and diffusive terms in $\boldsymbol{R}^n$ are discretized. The second order derivatives in the diffusive terms are discretized centrally.



In x-direction we first have

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

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

These two equations are used to form the discretized second order derivative :

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

The convective terms are treated with upwind discretization, which is controlled by an upwind parameter α.

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

$\alpha = 1$ corresponds to a fully upwind discretization. The same can be done for the convective and diffusive terms in y-direction and for v. Now the Poisson equation can be solved by an iterative process, which will be discussed in the next section.

3.3 Iteration - MILU

The Poisson equation can be schematically written as

$$Ax = b,$$

where

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

SAVOF96 uses a Modified Incomplete **LU** decomposition (MILU) to solve the above equation. MILU is a combination of the Conjugate Gradient method with a suitable preconditioner. The matrix A is decomposed in a lower triangular matrix L and an upper triangular matrix U. These matrices have 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 diagonals. The elements of these diagonals are called 'fill-in' elements. If these elements are ignored, it is possible to find a L and a U whose product equals A. Those ignored fill-in elements can result in unreliable solutions when these elements are rather big. To compensate this effect the fill-in is subtracted from the diagonal entries. With the mentioned preconditioner (K) the algorithm used for the conjugate gradient method becomes :

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

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

2) $x^{(n+1)} = x^{(n)} + \alpha_n z^{(n)}$ with $\alpha_n = (x^{(n)}, K^{-1}r^{(n)})/(z^{(n)}, Az^{(n)})$

3) $r^{(n+1)} = r^{(n)} - \alpha_n Az^{(n)}$

4) $z^{(n+1)} = K^{-1}r^{(n+1)} + \beta_n z^{(n)}$ with $\beta_n = (r^{(n+1)}, K^{-1}r^{(n+1)})/(z^{(n)}, K^{-1}z^{(n)})$

This algorithm has been implemented in the subroutine `MILU` and gives us p^{n+1}. The new velocity can be calculated 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

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

3.4 Description of a free surface

During calculation it is necessary to keep track of the free surface of a fluid. This is important, because the momentum equation has to be applied only in cells that contain an amount of fluid. SAVOF96 makes use of a so called *indicator function*. In section 2.3 the indicator

function was already mentioned. Now we will discuss the discretized version, which also will be named F.

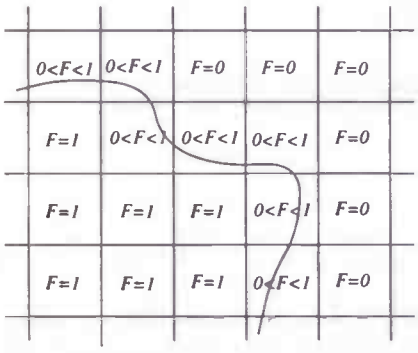


Figure 3.1: *free surface description.*

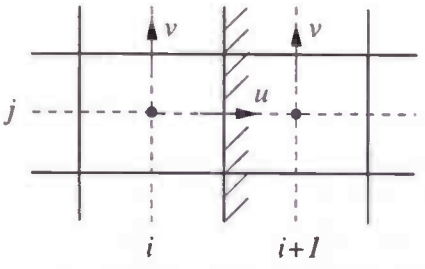
This indicator function is a function $F(i,j)$ that is equal to 1 if the cell (i,j) is completely filled with fluid and equal to 0 if it is empty. $F(i,j)$ can also have values between 0 and 1, depending on the percentage of the cell that is filled (see fig 3.1). There also exists a cell labelling, which gives more qualitative information about a cell. `NF(i,j)` is a two-dimensional array in which this information for cell `(i,j)` is stored. `NF(i,j)` can have the following values:

- 0: full cell,
- 1: surface cell with full cell to the left,
- 2: surface cell with full cell to the right,
- 3: surface cell with full cell at the bottom,
- 4: surface cell with full cell at the top,
- 5: degenerated cell,
- 6: empty cell,
- 7: 'outflow' cell,
- 8: 'inflow' cell,
- 9: obstacle or boundary cell.

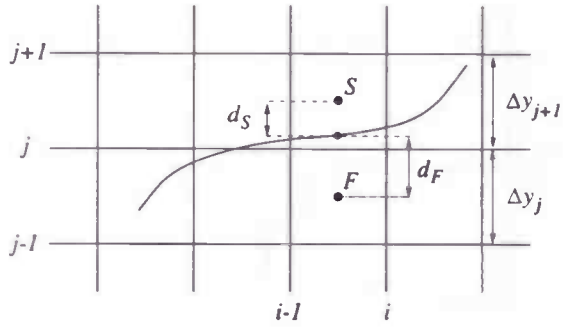
Using this labelling one knows where to apply the momentum equations and where the boundary conditions.

3.5 Boundary and inflow conditions

As noted in section 2.3, the boundary conditions for the Navier-Stokes equations usually are $u = v = 0$ on solid walls (no-slip condition). Since not all the velocities are defined on the wall because of the placement of the variables, we have to interpolate and make use of mirror points. For example, consider a vertical wall. The horizontal velocity u is defined on the wall, so that causes no problems. The vertical velocity v is not defined on the wall, but is placed at half a cell-distance. We set $v(i,j) = -v(i+1,j)$, so that after interpolation it is guaranteed that the vertical velocity is zero on the wall.



The boundary conditions at the free surface were already mentioned in paragraph 2.3.2. The curvature H in equation 2.6 is calculated using information from the indicator function. The pressure at the free surface needed in (2.6) is interpolated as follows. Suppose we have a full cell with pressure p_F and above it a partially filled cell with pressure p_S. We can estimate the average height of the surface in the upper cell to be $F_{i,j+1}\Delta y_{j+1}$. The distance between the centre of the upper cell and the free surface then becomes : $d_S = \frac{1}{2}\Delta y_{j+1} - F_{i,j+1}\Delta y_{j+1}$. In the same way we can describe the distance from the centre of the lower cell and the free surface: $d_F = \frac{1}{2}\Delta y_j + F_{i,j+1}\Delta y_{j+1}$.



Let the distance between the centres of the cells be denoted by $d = \frac{1}{2}\Delta y_j + \frac{1}{2}\Delta y_{j+1}$. We can now linearly interpolate the pressure in the two cells to obtain the pressure at the free surface:

$$p_f = \frac{d_S p_F + d_F p_S}{d}.$$

The cells with the previously introduced label `NF = 8` form the inflow openings. These cells

are filled with fluid ($F_{i,j} = 1$) and the velocity is set at a prescribed value that depends on the orientation of the outer wall in which the opening lies. The inflow cells are further treated the same as boundary cells.

Chapter 4

Results

4.1 Introduction

In the previous two chapters the mathematical and numerical model were discussed. In this chapter the actual calculations that were performed will be discussed. First we want to determine whether the program works properly. Therefore the draining of a cylindrical tank through a hole in the centre of the bottom was simulated and the results were compared to those found by Lubin and Springer. After that, another phenomenon that occurred is discussed. Then a closer look is taken at a more realistic situation, which is the quasi-steady flow. Here inflow areas were created, what can be compared to oil flowing over the bulkheads. These results will be used to derive a formula for the critical height.

4.2 Test-case

In order to test the validity of the program, some results of simulations were compared to literature on this topic.

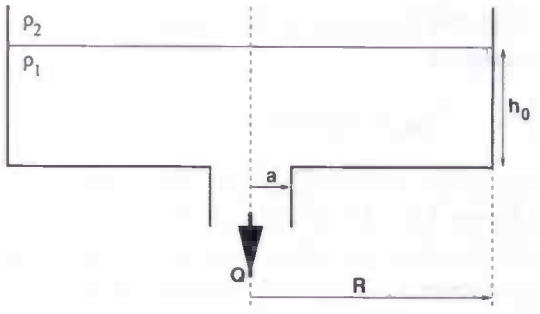


Figure 4.1: *A tank filled with fluid.*

In figure 4.1, we see a tank filled with fluid. This tank represents one compartment of a sump. a is the radius of the drain, Q the volume flow rate, h_0 the initial height and R is the radius of the tank. (So $2R$ represents the distance between two bulkheads). The outflow is directed downwards here, while in reality it is directed side-wards. The formation of a dip on the surface of an initially stationary liquid draining from a cylindrical tank was studied

experimentally and analytically by B.T. Lubin and G.S. Springer. Based upon theory, they derived an analytic formula predicting the height of the liquid surface at which the dip forms (H_c denotes the critical height) :

$$\frac{H_c}{a} = 0.69 \cdot \left[\frac{Q^2}{(1-\rho_2/\rho_1)ga^5} \right]^{1/5}, \tag{4.1}$$

where g is the gravitational acceleration, ρ_1 is the density of the bottom fluid and ρ_2 is the density of the top fluid. Lubin and Springer observed that the formation of a dip is so quick that it appears to extend into the drain almost instantly. The height at which a dip forms and then extends almost instantly into the drain is defined as the critical height H_c. If at a certain initial height the dip does not extend into the drain immediately, the critical height has not yet been reached. For later use, it is convenient to discuss the derivation of this formula. Lubin and Springer based their analysis on the assumption of quasi-steady flow in order to predict the height at which a dip forms. Furthermore they assumed that the effects of viscosity of both fluids is negligible.

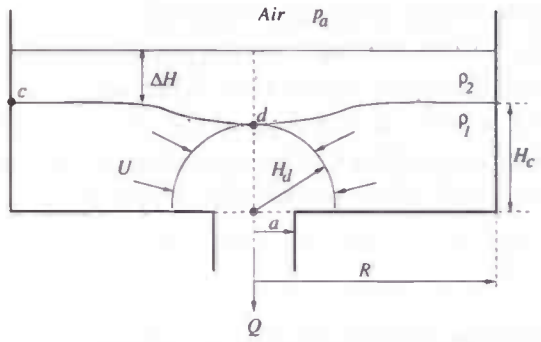


Figure 4.2: *Definitions of symbols used in analysis.*

Lubin and Springer stated that the pressure at the interface of the top and bottom fluids is due to hydrostatic pressure only, i.e.

$$p_{if} = \rho_2 g \Delta H + p_a, \tag{4.2}$$

where p_{if} is the pressure at any point on the interface. The other symbols and those mentioned further on are identified in figure 4.2. To determine the volume flow rate Q, they assumed that at the instant of the formation of the initial depression the conservation of mass for the hemispherical control volume shown in figure 4.2 may be expressed as

$$Q = 2\pi U_d H_d^2. \tag{4.3}$$

The steady Bernoulli equation along a streamline just below the interface is :

$$p_c + \rho_1 g H_c + \frac{1}{2}\rho_1 U_c^2 = p_d + \rho_1 g H_d + \frac{1}{2}\rho_1 U_d^2. \tag{4.4}$$

It is assumed that $U_c = 0$. In writing (4.3) and (4.4) it was further assumed that at every point at the surface of the control volume the flow velocity has the same value and is normal

to its surface. Equations (4.2)-(4.4) can be rearranged to yield

$$H_c = H_d + \frac{Q^2}{8\pi^2(1-\rho_2/\rho_1)gH_d^4}. \tag{4.5}$$

The experimental observation that the dip grows very rapidly is used to eliminate H_d by stating that at the instant of dip formation

$$\frac{dH_c}{dt} / \frac{dH_d}{dt} \simeq 0. \tag{4.6}$$

Combining the last two equations leads to the above mentioned formula.

For these simulations the bottom fluid is water and no top fluid is used. Therefore ρ_2 is the density of the air above the water. If we substitute the definition of the Froude number, $F = \frac{Q}{a^2 \cdot (ga)^{1/2}}$, and use the fact that $\rho_2 \approx 0$, we can rewrite (4.1) into the following form :

$$\frac{H_c}{a} = 0.69 \cdot F^{2/5}. \tag{4.7}$$

In figure 4.4, the lower line represents this formula and the upper line corresponds to the simulations. Figure 4.3 shows us the results of the refinement of the grid. The critical heights were calculated for a grid of 64×64 cells an for one of 128×128 cells. The method of determining the critical heights is the same as that mentioned before. The difference between the consecutive initial heights is about 3 mm. One can clearly see that the results of the simulations are the same, but it is possible that within these 3 mm the critical heights differ a little. This means that calculations with a grid of 64×64 cells have (almost) the same solution as calculations with a grid of 128×128 cells. So now we have solutions which are independent of the grid size and that is exactly what we want.

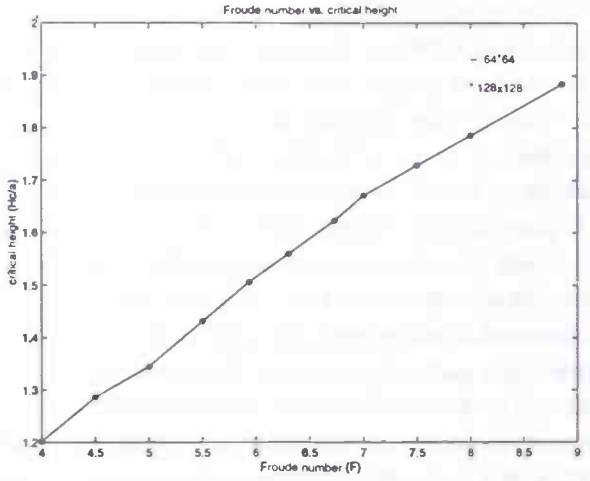


Figure 4.3: *Grid refinement.*

Lubin and Springer based their theory on potential flow, whereas we use the Navier-Stokes

equations. Solving the unsteady inviscid Navier-Stokes equations for irrotational flows is equivalent to solving the unsteady Bernoulli equation. However, Lubin and Springer apply the steady Bernoulli equation. We can choose the kinematic viscosity of water equal to zero, but the equations remain unsteady. The unsteady part of the Navier-Stokes equations ($\frac{\partial \boldsymbol{u}}{\partial t}$) is responsible for the difference between the two graphs. Large Froude numbers correspond to large drain velocities and consequently larger values of $\frac{\partial \boldsymbol{u}}{\partial t}$. This explains the increase of the difference between the graphs. Despite of the slight difference in the graph we can state that the program works properly.

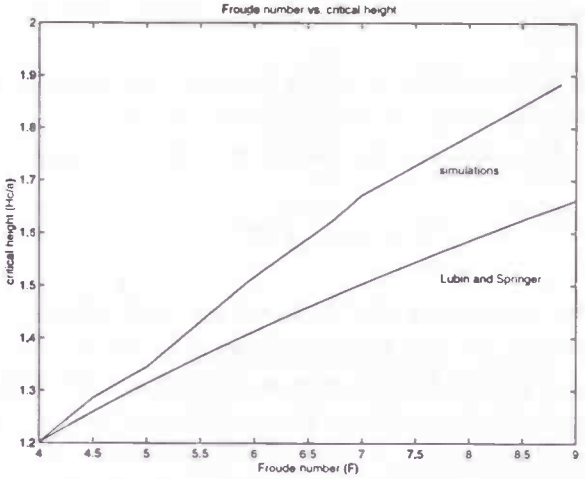


Figure 4.4: *The critical height for different Froude numbers.*

We will now discuss the method of determining the critical heights. At first, an initial guess based upon formula 4.7 is made for the initial height. If this does not result in the formation of a dip, the initial height is taken a bit lower. This continues until a dip is formed instantly. The critical height is therefore equal to the initial one. If the initial height is taken above the critical one, a dip is formed not before the fluid has been drained almost entirely out of the tank and long after the critical height has been reached. We will discuss this issue in more detail in section 4.4. The critical heights increase with the Froude numbers. In these simulations, Q is set to a constant value of 0.0347 m^3/s. So the only parameter that is not a constant is the drain radius a. Using this, we can conclude from figure 4.4 that a smaller drain radius results in a higher critical height and vice versa. This will be explained using the following example: Suppose two simulations are performed. In the first case the drain radius is a and in the second case it is $3a$. In the first case the maximum draw-down velocity occurring at the centre of the sink is nine times as large as that in the second case. Therefore a dip is formed sooner if the drain radius is smaller. The formation of a dip is shown in figure 4.5. The time development of the free surface can be seen in figure 4.6.
The critical heights obtained from the simulations are equal to the initial heights. If the initial heights are larger than the critical ones, strong free-surface oscillations occur and no dip is formed when expected. Lubin and Springer assumed that the flow was inviscid and quasi-steady and therefore applied the steady version of the Bernoulli equation:

$$p + \frac{1}{2}\rho U^2 + \rho g H = constant.$$

Here, p is the local fluid pressure, g is the acceleration due to gravity, H is a vertical coordinate measured from some convenient reference point and U is the flow velocity.

Figure 4.5: *formation of a dip.*

Bernoulli's equation simply states that the change in kinetic energy along any streamline is due to the work done by gravity and pressure. This is a way of stating that the mechanical energy of a small fluid element is conserved when there is no friction. In the simulations we have some form of friction, namely viscosity. Furthermore, U_c in equation 4.4 is not equal to zero, so it can not be ignored. This is the reason why no dip is formed when the initial heights are larger than the critical heights.

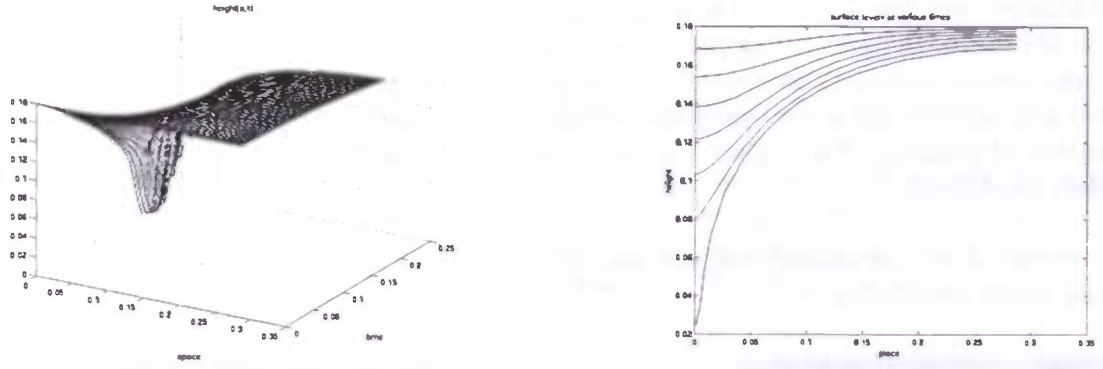


Figure 4.6: *Time development of the free surface.*

As mentioned earlier, the critical heights were computed for a constant volume flow rate and a variable drain radius. Our goal is to derive a simple formula for the critical height. Therefore we also need to compute critical heights for different volume flow rates. We want to examine whether there is a difference between the critical heights found through varying Q and those found through varying a. The results of these simulations are shown in table 4.1. One can see that there is no difference between the two types of simulations. This is desirable because it is now possible to derive a formula that is valid for all Q and a. From this table we can also conclude that a larger volume flow rate results in a larger critical height and vice versa.

Table 4.1 : Critical heights for different Froude numbers

Froude number	drain radius (m)	drain rate (m^3/s)	velocity (m/s)	critical height
4.0	0.095	0.0347	1.22	1.202
	0.081	0.0234	1.14	1.202
4.5	0.091	0.0347	1.33	1.271
	0.081	0.0263	1.28	1.271
5.0	0.087	0.0347	1.46	1.345
	0.081	0.0292	1.42	1.345
5.9	0.081	0.0347	1.68	1.506
	0.081	0.0347	1.68	1.506
6.7	0.077	0.0347	1.86	1.623
	0.081	0.0394	1.91	1.623
7.0	0.076	0.0347	1.91	1.671
	0.081	0.0407	1.97	1.671
8.9	0.069	0.0347	2.32	1.884
	0.081	0.0518	2.51	1.884

4.3 Formation of a jet

The formation of a dip is not the only phenomenon we encountered. If we choose the initial height just above the critical height another phenomenon occurs. At first, a dip seems to form but suddenly a jet appears in the centre of the depression region. This can be seen in figure 4.7. The formation of a jet is caused by the off-centre surface particles squeezing those fluid particles near the centre of the surface and forcing them to move upwards instead of down to the drain. In figure 4.8 we can clearly see that the fluid moves upwards in the middle of the depression region. The formation of a jet was also observed by Q. Zhou and W.P. Graebel who numerically investigated the withdrawal of a fluid from an open tank under the assumption of potential flow. They used a boundary-integral-method scheme with built-in boundary conditions.

Their numerical results showed two different phenomena, depending upon the volume flow rate and initial conditions.

Figure 4.7: *Simulation of a jet.*

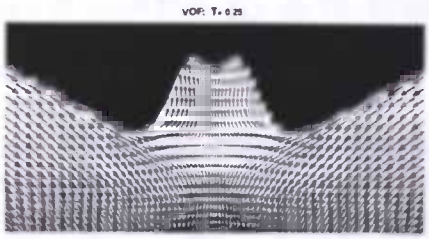


Figure 4.8: *Velocity vectors.*

When the tank is rapidly drained, a dip forms at the centre of the free surface. For a slowly draining tank, a jet forms in the centre of the depression region. These observations can be used to our advantage. As we have seen before, a larger volume flow rate will result in a larger critical height and consequently it will result in a larger height at which a jet is formed. Zhou and Graebel also found that the drain radius has an effect on whether a dip is formed

or a jet. If a certain initial height gives rise to the formation of a jet, a smaller drain radius may have the result that instead of a jet a dip is formed.
For certain initial heights a jet forms, but for small initial heights and small Froude numbers, no dip or jet is formed at any time. The free surface is so close to the drain that the fluid above the drain pipe flows directly into the drain. The areas with all the occurring phenomena are shown in figure 4.9. The line separating the 'dip' area and the 'no dip/jet' area is only a rough estimation. It is quite difficult to determine the exact border.

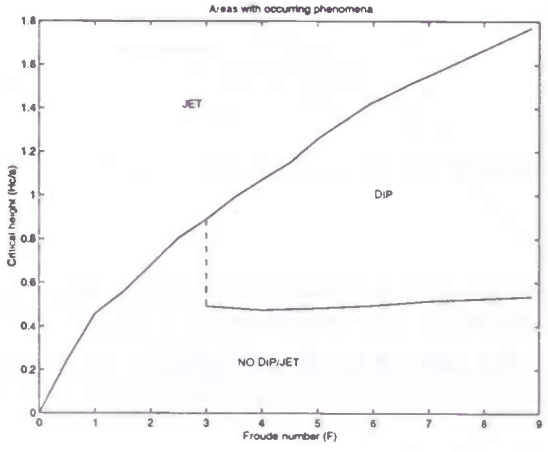


Figure 4.9: *Different phenomena.*

4.4 Quasi-steady flow

After the simulations done in section 4.1 a new method was used to determine the critical height. We want the free surface to drop very slowly and avoid too much surface oscillations in order to simulate a quasi-steady flow. This is done by creating an inflow area on the boundary of the tank (see figure 4.10). The inflow rate Q_{in} should be almost equal to the

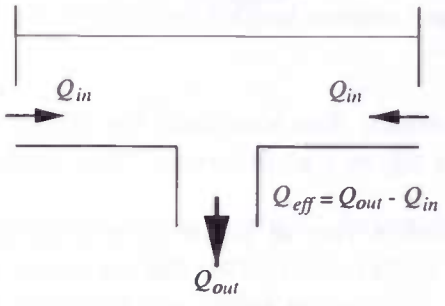


Figure 4.10: *Quasi-steady flow.*

volume flow rate Q_{out}. This ensures us that the free surface drops very slowly. It is expected that the critical heights found are smaller than those found in section 4.1 (see figure 4.11).

Quasi-steady flow is more like the real situation in an oilsump. The oil flows over and through the bulkheads. So this can be seen as some sort of inflow area.

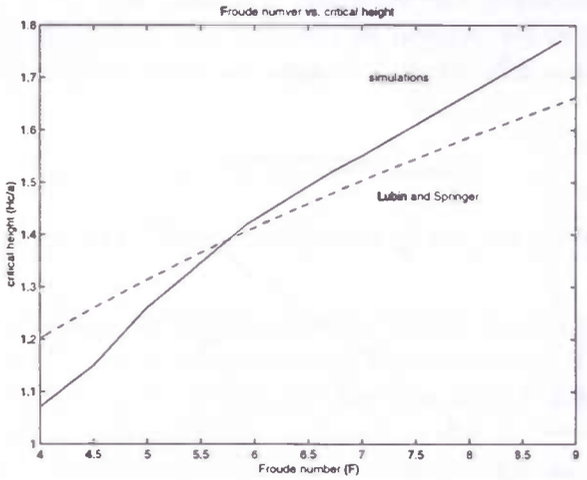


Figure 4.11: *The critical height for different Froude numbers.*

First simulations were performed using an inflow rate of 95% of the volume flow rate. The critical heights were determined in the same way as in section 4.1. This means that the initial heights are chosen in such a way that a dip is formed instantly (initial height = critical height). The results of these simulations are shown in figure 4.11. We can see that the computed critical heights are somewhat lower compared to those found in section 4.1. Furthermore, until $F \approx 5.75$ the line is situated underneath the formula of Lubin and Springer. So if the Froude number is not bigger than about 5.75 it is safe to choose the initial height of the oil according to the formula of Lubin and Springer. In these simulations the kinematic viscosity was set to 3.2e-5 m^2/s what corresponds with an oil temperature of 75° C. Simulations with $\nu = 5.0\text{e-}4$ m^2/s resulted in practically the same critical heights. This kinematic viscosity corresponds to a temperature of 40° C. This means that the viscosity of the oil has a negligible influence on the critical height. We can conclude that the inflow has a positive influence on the height at which a dip is formed. Lower critical heights result in lower initial heights that can be chosen.

The idea of simulating a quasi-steady flow is to drain the oil slowly from an arbitrary initial height and then look whether a dip or a jet is formed. This is what is done next.

Simulations were performed with a rate of inflow of 80% of the volume flow rate, $h_0 = 0.19\ m$, $a = 0.081\ m$ and $Q = 0.0347\ m^3/s$. This did not result in a quasi-steady flow. The free surface oscillated heavily. The rate of inflow was increased up to 99.5% of the volume flow rate. After a short period of heavy oscillations the free surface became smooth and the fluid flowed slowly down the drain. At a certain point in time a jet formed in the centre of the free surface. For other initial heights a jet was also formed.

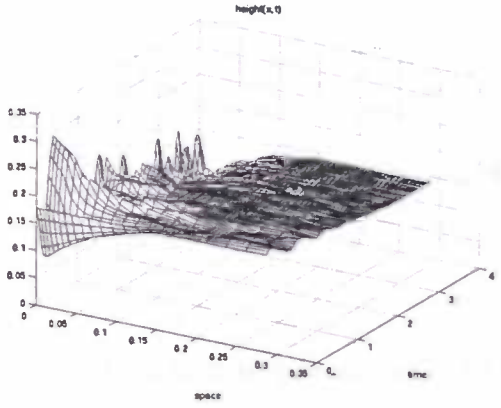


Figure 4.12: $Q_{in} = 0.8 \times Q_{out}$, *not yet quasi-steady.*

Another method was used in an attempt to simulate a dip. First an inflow of 99.5% of the volume flow rate was used to create a quasi-steady flow as shown in figure 4.14. Then the rate of inflow was lowered to 90% of Q. This was done using the restart file `savof96.rst` (see section A.3). Immediately after restarting the simulation, a jet forms in the centre of the free surface.
This procedure was repeated several times, each time with different rates of inflow (after the restart), but with the same result.

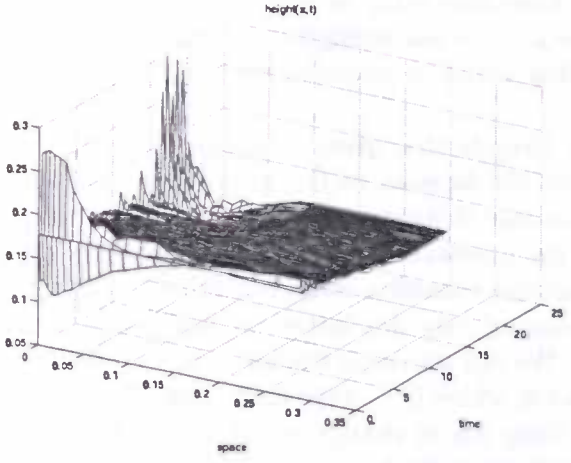


Figure 4.13: *Quasi-steady flow with formation of a jet.*

For evaluation purposes simulations were performed with $a = 0.069\ m$ and $a = 0.095\ m$. Using these drain radii resulted also in the formation of a jet. So all these simulations lead to the formation of a jet. This is similar to the results of simulations done without creating a quasi-steady flow and choosing the initial height above the critical one. Comparing all the

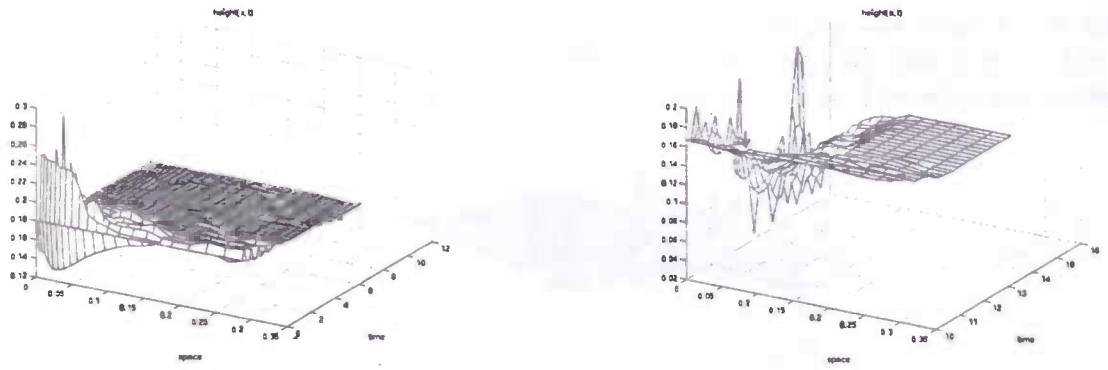


Figure 4.14: *Time development of the free surface.*

results shows us that movement of the free surface before the critical height is reached has a big influence on whether a dip is formed or not.This can be explained using the Bernoulli equation. In deriving their formula, Lubin and Springer stated that the velocity of the free surface at the end of tank (U_c) is negligible compared to that in the middle of the free surface (U_d) and is therefore set to zero (see section 4.2). If we perform simulations choosing $h_0 = H_c$ then $U_c = 0$ at the beginning of the simulation and a dip is formed instantly. If $h_0 > H_c$ $U_c \neq 0$ when the free surface is in the neighbourhood of H_c and consequently a jet is formed. Furthermore if we simulate a quasi-steady state U_c is not negligible compared to U_d. Using U_c in the Bernoulli equation the formula of Lubin and Springer changes into:

$$\frac{H_c}{a} = 0.69 \cdot \left[\frac{{Q_{real}}^2 - {Q_{U_c}}^2}{(1-\rho_2/\rho_1)ga^5} \right]^{1/5}, \tag{4.8}$$

with Q_{real} as the "old" volume flow rate. In deriving this adjusted formula it was assumed that Q_{U_c} is a constant. So if U_c is not negligible compared to U_d it results in lower critical heights and that is consistent with the results of the simulations.

We now want to derive a formula that gives an accurate approximation of the line in figure 4.9. For this we adjust the formula of Lubin and Springer. The exponent in formula 4.7 must not be altered, because this exponent comes from dimensional analysis (see Froude number). We can change the constant factor 0.69. This factor is determined by the volume flow rate Q, and for Q a control volume is chosen. The usage of this control volume is somewhat questionable. By changing only this factor, we will not be able to acquire an accurate approximation of the line. We can also shift the formula a bit to the right. This corresponds to what is done in formula 4.8, where Q_{U_c} is the shift. Here $Q_{U_c} = 0$ so we should not use this either. But it is the only thing left to change, which means that we have to use a shift. By using a shift the formula will not pass the origin. This does not cause any problem, because the formula we derive is only valid for Froude numbers in the range 4 to 9. Variations of these two components lead to the following approximation of the results of the simulations:

$$\frac{H_c}{a} = 0.76 \cdot (F^2 - 11.5)^{1/5}. \tag{4.9}$$

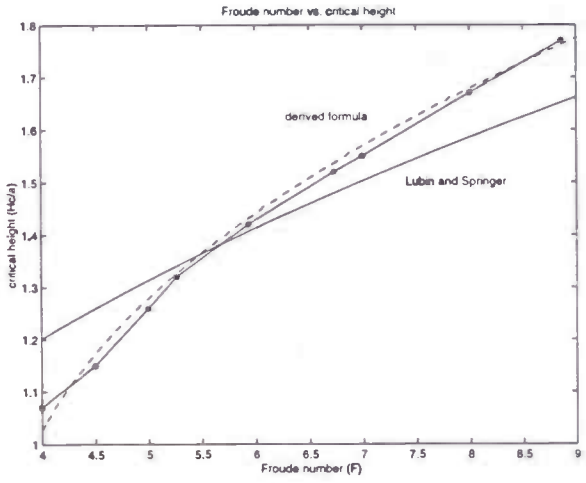


Figure 4.15: *Approximation of the results.*

The following figure shows the pipe diameter versus the volume flow rate. This can be used to determine the pipe diameter necessary for a certain flow rate and flow velocity. The next figure can then be used to see what the critical height is for a certain flow rate and pipe diameter.

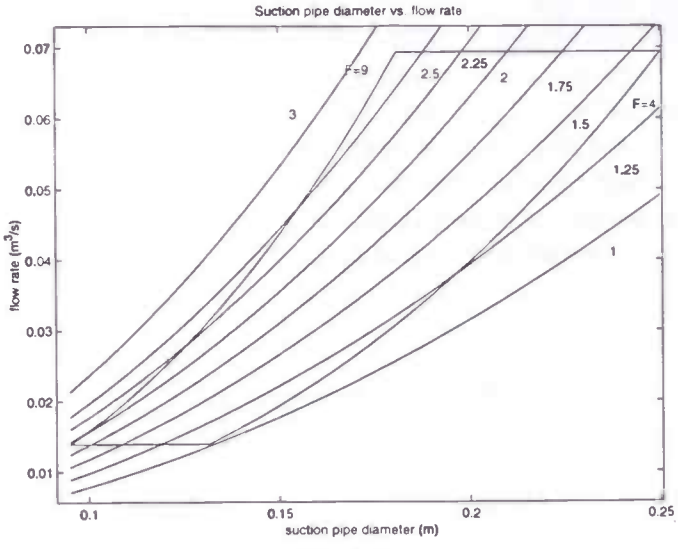


Figure 4.16: *pipe diameter vs. flow rate*

Because of the fact that formula 4.9 is only valid for Froude numbers in the range 4 to 9, this area was also indicated in figure 4.16. The curves denote the flow velocities in m/s. For a certain flow rate and pipe diameter the following figure can be used to determine the critical height.

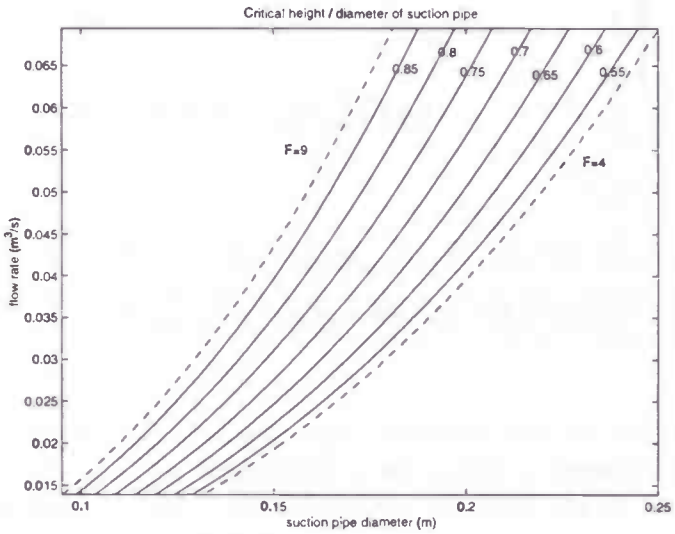


Figure 4.17: *critical height*

The curves in the figure denote the critical heights divided by the diameter of the suction pipe.

Chapter 5

Conclusions

The purpose of this report was to derive a simple formula for the critical height containing all the important parameters. First of all, the formula of Lubin and Springer gives quite a good indication of the critical height and the parameters that influence it. It is clear that the most important parameters are the drain radius a and the volume flow rate Q. Smaller drain radii result in higher critical heights and higher volume flow rates result also in higher critical heights. The simulations that were performed, resulted in critical heights in the neighbourhood of the formula of Lubin and Springer, but only when the initial heights are equal to the critical ones. Furthermore we can conclude that an amount of inflow has a positive influence on the critical heights. This results in lower heights at which the dip is formed. In the literature, all theories are based upon potential flow and the assumption of a quasi-steady situation was made. In reality the flow is viscous and the assumption of a quasi-steady flow is questionable. Therefore the theory of Lubin and Springer has only limited value. In our model movement of the free surface before the critical height is reached does not result in the formation of a dip. Therefore a different method of determining the critical height was used. The flow must be started with the initial height equal to or less than the critical one, otherwise a jet is formed. Of course, a dip is always formed when the tank is almost entirely empty. After having done all the simulations, we can conclude that the definition of the critical height that was used was probably not entirely correct. Based upon the simulations performed the following formula for the critical height was derived, for Froude numbers in the range 4 to 9 :

$$\frac{H_c}{a} = 0.76 \cdot (F^2 - 11.5)^{1/5},$$

where $F = \frac{Q}{(a^2(g \cdot a)^{1/2})}$.

Appendix A

Program description

The numerical model has been implemented in the program SAVOF96. This appendix gives more detailed information about the calling sequence and the several variables and subroutines. Furthermore, a brief explanation of the input and output files is given.

A.1 FORTAN 77

A.1.1 Calling sequence

The calling sequence can be represented as follows:

```
initalisation   SETPAR   GRID
                SETFLD   SURDEF
                         MKGEOM
                         RDGRID
                         LDSTAT
                         LABEL
                         BC       BCBND
time step       INIT     BCBND
                PETCAL
                TILDE    BDYFRC
                SOLVEP   COEFF
                         CONVRT
                         PRESIT   SLAG
                         MILU
                         BC       BCBND
                CFLCHK
                DTADJ
                VFCONV   LABEL
                SVSTAT
```

The post-processing routines were excluded for presentational reasons.

A.1.2 Common block variables

The variables in a *common block* can be used in every subroutine in which this common block is declared. This enables subroutines to exchange data. The important variables are mentioned below together with a short description.

/CASE/ contains external body forces and their controls:

Omega, DelOme	: rotation rate and initial rotation relative to Omega
XO, YO	: in the case of 2D rotation : point about which the rotation takes place
TwUp, TwDown	: time to start and end the rotation
UO, VO	: initial velocities in x- and y-direction
Ampl, Freq, AAngle	: amplitude, frequency and the angle under which an oscillation is formed
GX, GY	: accelerations in x- and y-direction
TxOn, TxOff, TyOn, TyOff	: times to turn accelerations in x- and y- direction on and off

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

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

/GRIDAR/ contains parameters involving grid size:

X(I)	: x-co-ordinates of the grid points
XI(I)	: x-co-ordinates of cell centres
DelX(I)	: distance in x-direction between two subsequent grid points ($\Delta x_i = x_i - x_{i-1}$)
Y(J)	: y-co-ordinates of the grid points
YI(J)	: y-co-ordinates of cell centres
DelY(J)	: distance in y-direction between two subsequent grid points ($\Delta y_i = y_i - y_{i-1}$)
RX(I),RXI(I), RDX(I)	: inverse of X(I), XI(I) and DelX(I)
Circum(I)	: the circumference of a circle with radius x_i (in the axisymmetrical case, 1 otherwise)
RDY(J)	: inverse of DelY(J)
CX, CY	: stretch parameters in x- and y-direction
CYL, ICYL	: floating-point and integer version of the 2D/ axisymmetric switch (CYLL=0 for 2D andCYLL=1 for axisymmetric geometries)
IMaxUs, JMaxUs, IM1Us, JM1Us, IM2Us,	: number of grid points in x- and y-direction (IMaxUs=IM1Us+1=IM2Us+2, JMaxUs=JM1Us+1=JM2Us+2)

JM2Us

/ORGA/ is used for cell labelling:

NF(I,J),	: contain cell labels for the current and
NFN(I,J)	previous time level respectively
PETA(I,J)	: coefficient used to interpolate the pressure

/PHYS/ contains pressure and velocities at the current and previous time level.

F(I,J), FN(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
PS(I,J)	: pressure at free surface
VMAX	: maximum attained velocity
PMIN, PMAX	: minimum and maximum pressure

VMAX, PMIN and PMAX are used for scaling during post-processing.

/TIMES/ contains parameters related to time levels and time 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

A short description of the subroutines is now given.

BC	: sets the boundary conditions for the velocity components
BCBND	: boundary conditions at outer boundary
BDYFRC	: computes the apparent body force
CFLCHK	: monitors the CFL-number and sets flag for time-step adjustment
COEFF	: defines the coefficient matrix for the Poisson equation including boundary conditions at the wall and free surface
CONVRT	: converts 2D data-structure into 1D data-structure for more efficient implementation of the pressure solver
DTADJ	: halves/doubles the time step (old time step will be repeated)
GRID	: makes a (non-uniform) grid
INIT	: starts a new time step
LABEL	: labels empty, surface (preliminary) and full cells
LDSTAT	: reads the restart file (produced by SVSTAT)
MILU	: solves Poisson equation trough MILU algorithm
MKGEOM	: defines the geometry

PETCAL	: labels surface cells and computes pressure at free surface with a local height function. Here the position of the free surface is calculated and that position will be used in COEFF to interpolate for the pressure boundary condition at the free surface
PRESIT	: solves Poisson equation and controls SOR relaxation factor
PRT	: prints and writes results
RDGRID	: reads 'broidary' file defining geometry
SETFLD	: initialises fluid configuration
SETPAR	: reads input file
SLAG	: performs one SOR sweep. It uses a 1D implementation
SOLVEP	: organises pressure calculation and updates velocity
SURDEF	: generates fluid configuration
TILDE	: integrates momentum equations
VFCONV	: moves fluid, i.e. adjusts VOF-function and re-labels
AVS	: generates output to be processed by AVS
GNUPLT	: writes data to a pipe to be displayed by GNUPLOT
FILOUT	: determines and saves fill ratios of specified regions
FLXOUT	: determines and saves fluxes trough specified areas
FRCOUT	: integrates the pressure on the walls in both x- and y-direction and writes the resulting force to file
MKSTRL	: keeps track of specified particles by integrating velocities
MPTOUT	: interpolates velocities in points specified in the geometry file and saves them
PRTFLD	: produces a sequence of characters representing the entire geometry including obstacles
SVSTAT	: saves crucial variables to file for later use

A.1.4 Program alterations

A few alterations have been made in SAVOF96 to make it more suitable for this particular problem. The following subroutines have been adjusted:

- SETPAR
  In this subroutine a line is added concerning the inflow velocities. It is now possible to prescribe inflow openigs on every boundary with different inflow velocities. This adjustment has also been made to the main input file (see A.2.1).

```
C**** BOUNDARY CONDITIONS AND IN-/OUTFLOW
      READ(5,*)
      READ(5,*)
      READ(5,*) WL, WR, WT, WB, UIn, VIn, FreqIn, IPIn, PIn
      READ(5,*)
      READ(5,*) UWIn, VWIn, UOIn, VOIn, UNIn, VNIn
      READ(5,*)
```

- BCBND
  The inflow velocities implemented in subroutine SETPAR are used here for the boundary conditions on the outer walls. For example, the horizontal velocity UIn on the western boundary has been replaced by UWIn, etc.

```
c*** vertical walls

      DO 100 J=1,JMaxUs
c-- western boundary
         V(1,J)=SgnL*V(2,J)
         W(1,J)=SgnLW*W(2,J)

         IF (nf(1,j).EQ.9) THEN
            U(1,J)=0.0
            P(1,J)=P(2,J)
         ELSEIF (nf(1,j).EQ.8) THEN
            u(1,j)=UWIn*COS(FreqIn*t)
            u(0,j)=u(1,j)
            v(1,j)=2.*VWIn*COS(FreqIn*t) - v(2,j)
            P(1,J)=P(2,J)
         ELSEIF (nf(1,j).EQ.7) THEN
            IF(nf(2,j).NE.0) u(1,j)=u(2,j)
            u(0,j)=2.0*u(1,j)-u(2,j)
c... negative outflow
            IF (u(1,j).GT.0.0) u(0,j)=u(1,j)
            P(1,J)=POutW
         ENDIF
```

  The same was done for the other boundaries, but this was left out for presentational reasons.

- PRT
  In this subroutine an adjustment has been made to determine the height of the free surface at several moments in time. The results are written to the file **height.res**.

```
c--- height in each column

         DO 210 I=2,IM1Us
            HOOG=0.0
            DO 209 J=2,JM1Us
               STEP=F(I,J)
               IF (NF(I,J).EQ.9) STEP=1.0
  209       HOOG=HOOG+STEP*DelY(J)
  210    HGT(I)=HOOG
```

A.2 Input files

A.2.1 Main input

SAVOF96 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 looks like this:

```
SAVOF98-2.0
***** tank geometry ************************************************
iCyl      Xmin       Xmax      Ymin      Ymax      SpGeom      SpMot
 1         0.0        0.3       0.0       0.3         1          0

***** liquid configuration *****************************************
LiqCnf     xp     yp      r      xq      yq
  3       0.0    0.0    0.0     0.3     0.18

***** grid definition **********************************************
iMaxUs    jMaxUs     cx     cy     Xpos     Ypos
  64        64        1      1      0.0     0.06

***** liquid properties ********************************************
Sigma      CAngle        Nu
 0.0        90.0      3.2e-5

***** 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      90.0
Rpm      DelOme     TwUp     TwDown     x0      y0
0.0        0.0       0.0      100.0    0.0     0.0

***** body forces and external motion: axisymmetric *****************
 Gy      TyOn     TyOff     v0      Ampl      Freq
-9.8      0.0     100.0    0.0       0.0       0.0
 Rpm     DelOme    TwUp    TwDown
 0.0      0.0       0.0      0.0

***** boundary conditions and inflow characteristics ***************
left     right     top    bottom    UZIn    VZIn    Freqin   IPIN    PIn
 2         2        1        2       0.0    -1.68      0.0      0     0.0
UWIn     VWIn     UOIn     VOIn     VNIn     VNIn
0.0       0.0      0.0      0.0      0.0      0.0

***** upwind parameter and Poisson iteration parameters  *************
Alpha     Epsi      ItMax      OmStrt      IMilu
1.0     1.0e-4        50         1.9          1
```

```
***** time step and restart control *********************************
TFin      DelT      CFLMax      PrtDt      svst     svdt
1.0       5e-5        0.6        0.01        1       0.1

***** print / plot control *****************************************
gnu     avs    uvpf    velop    height    force
 0       0       1       0        1         0

***** stream lines *************************************************
nrx     nry     nrdt     xps     yps    xqs    yqs     t     dt/delt
 0       0        0      0.0     0.0    0.0    0.0    0.0       0

***** fluxes *******************************************************
number of fluxes to be printed (flux01.out...flux**.out are created)
  1
  p1      p2      p3     hor  (line from (p1,p3) to (p2,p3) if hor=1 )
 0.0   0.081   0.060      1

***** fill ratios **************************************************
number of ratios to be printed (fill01.out...fill**.out are created)
   0
  xpf    ypf     xqf     yqf (box with nodes (xpf,ypf) and (xqf,yqf))
```

The above example will be used to explain the main input file. In the *tank geometry* section the following parameters have to be set. `icyl` is the switch between two-dimensional and axisymmetric geometries. (0 for two-dimensional and 1 for axisymmetric calculations). `(Xmin,Ymin)` and `(Xmax,Ymax)` are the lower left and upper right corner of the tank. (The same units should be used for all parameters.) If a more complex geometry is required, `SpGeom` must be set to 1. This geometry has to be defined in the file `savof96.geo` (see next section).

In the *liquid configuration* section the initial form and position of the fluid has to be specified. `LiqCnf` can have the following values :

1 : lower part of cylinder with the surface shaped as a semi-circle at average height `yp`;
2 : toroid (drop) with centre `(xp,yp)`, radius `r`;
3 : rectangle with nodes `(xp,yp)` and `(xp,yp)` ;
4 : drop along axis (y=yp, `r`) falling in water-pool deep `yq`;
5 : fluid filament with width `r`and height `yp` semi-sphere

The grid is defined in the section *grid definition*. `iMaxUs` and `jMaxUs` are the number of grid points in x- and y-direction respectively. `cx` and `cy` are the stretch parameters which determine the stretching from position `(Xpos,Ypos)`. When `cx` (`cy`)is set to 0 no stretching is performed. Positive values result in smaller cells near `Xpos` (`Ypos`) and negative values result in smaller cells away from `Xpos` (`Ypos`).

The *liquid properties* that can be defined are `Sigma` (kinematic surface tension $\frac{\sigma}{\rho}$), `CAngle` (contact angle) and `Nu` (kinematic viscosity $\frac{\mu}{\rho}$).

For the next two sections we refer to section A.2 where in the common block `CASE` all the parameters are explained.

Specified next are the *boundary conditions and inflow characteristics*. For every side of the tank, the desired boundary condition can be stated. Values `0` and `1` represent *slip* and *no-slip* boundary conditions respectively. Value `7` makes the entire side an opening for outflow and value `8` makes it an opening for inflow. Inflow can be specified by the velocity parameters. For example `UZIn` and `VZIn` are the inflow velocities on the northern boundary in x- and y-direction respectively. The inflow can be made to oscillate with the frequency `Freqin`. In the next line the inflow velocities on the western, eastern and southern boundary can be stated.

The numerical parameters are defined next. `Alpha` is the upwind parameter, which should be set to `0` for central and `1` for upwind discretization. With `Epsi`, the Poisson convergence can be controlled. `ItMax` is the maximum number of iterations that SAVOF96 is allowed to perform in one time step. `IMilu` is the switch between the Poisson solvers MILU (`1`) and SOR (`0`). If a SOR iteration is used, the initial relaxation factor can be prescribed using `OmStrt`.

In the section *time step and restart control* the end time `TFin` and the initial timestep `Delt` can be specified. The latter may be reduced or doubled by SAVOF96 if necessary and possible. `CFLMax` denotes the maximum allowed CFL number. Also embedded in this section is the control of the frequency at which the output is written to file. `PrtDt` is the time between two consecutive small 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. The last to parameters in this section are used to create 'back-ups'. `svst` can have three values: `0` if no restart backups are required, `1` to save the program state every `svst` time units and `2` is a saved state from a previous run is to be read at startup, and proceeds then like `1`.

The last four sections of the input file are devoted to SAVOF96's output. First we have the *print/plot control* section. It consists of six switches used to choose the kind of output SAVOF96 should produce. All the switches should be either `1`, to enable, or `0` to disable the output option. the following abbreviations appear:

`uvpf`	: velocity and pressure data for visualisation with MATLAB
`gnu`	: interactive visualisation through GNUPLOT
`avs`	: velocity and pressure data in AVS format
`velop`	: additional velocity and pressure output in `savof96.out`
`height`	: height of free surface at several moments in time
`force`	: x- and y-components of force exerted by liquid

Furthermore one can follow the path of a fluid particle in a fluid. This can be done in *streamlines*. The only thing that has to be taken care of here is that all the values are set to zero.

Next is the *fluxes* section. First, the number of fluxes to measure has to be specified. Then for all of the fluxes a line is to be added to the input file specifying the locating where to record the fluxes. Those locations are defined by horizontal and vertical lines. If `hor` is set to `1` then the line has start point (`p1,p3`) and end point (`p2,p3`), which results in a horizontal line. If `hor` is set to `0` the co-ordinates are (`p3,p1`) and (`p3,p2`) a vertical line is created. The resulting fluxes are written to `flux##.out`.

The *fill ratios* of a number of a rectangular areas can be monitored and written to file. It is sufficient to set the *number or ratios to be printed* to `0`.

A.2.2 Special geometry

The following file `savof96.geo` was used to create the oilsump.

```
OILSUMP
Icyl    XMin    XMax    Ymin    YMax
  1     0.0     0.3    0.0      0.3
cirkelbogen  lijnstukken  instromen  uitstromen
     0            1           2          0
cirkelbogen:
  xm     ym       r   teta1   teta2  kant links rechts boven onder
lijnstukken:
  x1       y1     x2      y2   kant links rechts boven onder meetptn
  0.081   0.0    0.3    0.06    2     0     1      0     0      0
instromen:
  p1     p2   kant (links=1,rechts=2,boven=3,onder=4)
0.0     0.081  4
0.061   0.111  2
uitstromen:
  p1     p2  kant (links=1,rechts=2,boven=3,onder=4)
```

In the first input line the parameter for two-dimensional (`Icyl=0`) and axisymmetric(`Icyl=1`) calculation is set and the dimensions of the geometry are given. To create all kinds of geometries one can use `cirkelbogen`circular arcs), `lijnstukken` (lines), `instromen` (inflow) and `uitstromen` (outflow). The number of those objects used are specified in the next line. Next, all the objects can be defined specifically. First the circular arcs are defined. (`xm,ym`) denotes the centre of such an arc and `r` the radius. `teta1` and `teta2` are the start and end angle 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 centre of the circle, and `1` if its complement has to be filled. The next four arguments tell SAVOF96 towards which outer wall the object is to be extended (either`0` or `1`). Every line of the '`cirkelbogen`' section represents an arc. Secondly lines (or rectangles) can be created. The are defined by their begin (`x1,y1`) and end (`x2,y2`) (or by their opposite corners (`x1,y1`) and (`x2,y2`)). The '`kant`' 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 (or - for vertical lines - to the right of) the line,
- 2 : results in a rectangle.

The other arguments have the same effect as they have for circular arcs. The last argument in the kant section gives a possibility to create extra information about the velocity nearby an object. Finally, the in- and outflow openings have to be stated. These openings can only be situated on the outer walls. Three input arguments are sufficient to describe in- and outflow openings: p1 and p2 determine the co-ordinates of the opening and kant is here used to determine the side at which the opening should be created (1 = left, 2 = right, 3 = upper and 4 = lower).

A.3 Output files

In this section the output files that were used will be discussed.

`savof96.out`
This is SAVOF96's main output file. It contains co-ordinates of the grid points, rough plots of the fluid inside the geometry and comments about the convergence, evolution of time-step, etc. It also contains lines with the points in time, followed by the number of iterations that was required to obtain the desired accuracy, the relative change in volume and an array representing the level of the fluid.

`grid.out`
This is a text file containing a representation of of the geometry and the fluid in it. The file is produced by the subroutine PRTFLD (see section A.3) and contains characters like # for obstacle cells, * for cells with fluid and I or O for in- and outflow cells respectively.

`upvf###.dat`
These files contain information about the velocity, pressure and the viscosity.

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

`height.res`
This file contains information about the height of the free surface at several moments in time.

`savof96.rst`
This is SAVOF96's restart file. It gives us the option to perform a calculation that has the previously saved state an an initial configuration.

List of symbols

Below a list of the important symbols used in this report is given with a short description. For some variables a discrete version is given. The subscript $_{i,j}$ refers to the cell number and the superscript ^{n} denotes the point in time.

x, y, x_i, y_j	: co-ordinates in horizontal and vertical direction respectively
$\Delta x_i, \Delta y_j$	: distance between two consecutive co-ordinates in x- and y-direction respectively.
z, r	: axial and radial co-ordinates respectively
$u, v, u^n_{i,j}, , v^n_{i,j}$	: velocity in horizontal and vertical direction respectively, or velocity in radial and axial direction respectively
$p, p^n_{i,j}$	: pressure
ρ	: density
μ	: dynamic viscosity
ν	: kinematic viscosity ($\nu = \frac{\mu}{\rho}$)
a	: drain radius
g	: gravitational acceleration
Q	: volume flow rate
R	: radius of the tank
h_0	: initial height of the free surface
H_c	: critical height
F	: Froude number ($F = \frac{Q}{a^2(g \cdot a)^{1/2}}$)
U_c	: velocity of the free surface at the end of the tank
U_d	: velocity of the free surface in the middle of the tank

Bibliography

[1] B. de Groot. SAVOF96 - Simulation of free-surface liquid dynamics in moving complex geometries. Master's thesis. Rijksuniversiteit Groningen, 1996

[2] B.T. Lubin and G.S. Springer. The formation of a dip on the surface of a liquid draining from a tank. *Journal of Fluid Mechanics*, Volume 29, pages 385-390, 1967.

[3] Q. Zhou and W.P. Graebel. Axisymmetric draining of a cylindrical tank with a free surface. *Journal of Fluid Mechanics*, Volume 221, pages 511-532, 1991

[4] G.C. Hocking and J.M. Vanden-Broeck. Draining of a fluid of finite depth into a vertical slot. *Applied Mathematical Modelling*, Volume 21, pages 643-649, 1997.

[5] A.E.P. Veldman. *Numerieke Stromingsleer*. Rijksuniversiteit Groningen, March 1994. Lecture notes

[6] E.F.F. Botta. *Eindige Differentiemethoden*. Rijksuniversiteit Groningen, November 1992. Lecture notes

[7] H.W. Hoogstraten. *Stromingsleer*. Rijksuniversiteit Groningen, June 1992. Lecture Notes