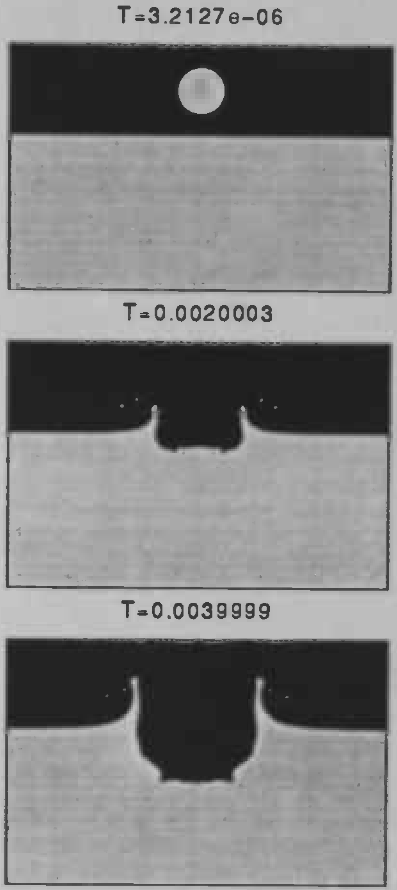
Simulation of Splashing and Splatting Droplets

Sushma Mewasingh



Department of
Mathematics

RuG

Master's thesis

Simulation of Splashing and Splatting Droplets

Sushma Mewasingh

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

April 1999

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 contains a report of my graduation work at the mathematics department of the RijksUniversiteit Groningen. I have been working on it under the supervision of Prof.dr.A.E.P.Veldman who supported me very well during this period. Therefor I would like to express my sincere gratitude.

In this paper a theoretical and numerical study of the deformation of a liquid droplet is presented.

Two cases have been studied namely:

1. the deformation of a liquid droplet colliding with a flat surface.
2. a droplet falling into a pool containing the same liquid.

Simulations were performed using the computer program SAVOF96.

Sushma Mewasingh , Groningen
April , 1999

Contents

Chapter 1

Introduction

Liquid droplets impacting on a surface are encountered in many practical processes such as spray cooling, spray forming, spray coating and inkjet-technology. A fundamental understanding of the fluid dynamics associated with droplet impingement is important for process development and further advancement. The complexity of the fluid dynamical aspects of droplet impingement on a surface is illustrated by the extreme deformation of the droplet surface ocurring within very short time scales.

To be able to solve any flow problem, the flow domain has to be subdivided into a number of volumes, or cells, to form a computational grid. After the grid generation the conditions at all boundaries of the grid must be supplied. The next step is to specify the properties of the fluid flowing through the domain. The partial differential equations describing the fluid flow are discretized over the mesh.

SAVOF96 is a fluid dynamics program for the solution of fluid flow. The program accounts for the presence of inertial, viscous, gravitational and surface tension effects. It solves the basic conservation equations of fluid dynamics and produces results in the form of color plots of velocities, pressures, temperatures and other flow variables. SAVOF96 is based on the finite difference method.

The fluid model presented here simulates the impact of a liquid droplet on a solid or liquid substrate - starting at the instant where the droplet comes into contact with the substrate - with a prescribed velocity. The model equations are listed first (Chapter 2) along with the appropriate boundary and initial conditions. The numerical solution procedure is then presented (Chapter 3) followed by some representative simulations (Chapter 4).

Chapter 2

Mathematical model

The physical aspects of any fluid flow are governed by the following fundamental principles :

- mass is conserved
- momentum is conserved

These basic principles can be expressed in the form of mathematical equations : **the Navier-Stokes equations**.

The mathematical model based on the complete Navier-Stokes equations is applied to the axisymmetric system shown in Fig.(2.1).

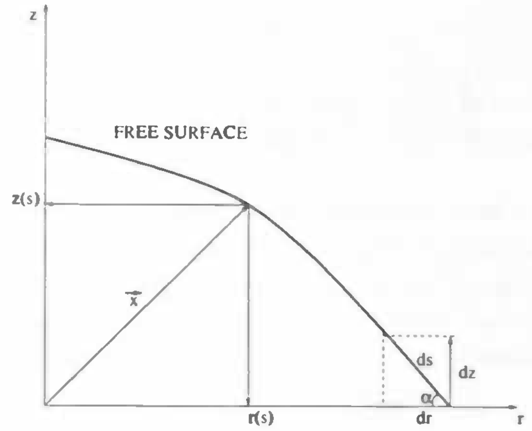


Figure 2.1: Scheme of the deforming droplet defining relevant notation.

This system represents the centerline cross section of a droplet spreading after impact on a flat surface.

In this case the following notation is used :

- r and z are respectively the radial and axial coordinates,

- $\mathbf{x}= \mathbf{x}[r(s),\ z(s)]$ is a position vector,
- s is the coordinate measured along the free surface,
- α is the angle between the solid surface and the free surface.

2.1 Navier-Stokes equations

Fluid motion can be described by the Navier-Stokes equations which consist of the two following 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 :

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

where

- $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ is the position,
- $\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$ is the velocity,
- t is the time,
- $p(x,y)$ is the pressure,
- F_x and F_y are the x-and y-components of an external force,
- $\rho(x,y)$ is the density,
- $\sigma = (\sigma_{ij})$ 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 :

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

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

$$\begin{aligned}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \\
\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= F_x - \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\
\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= F_y - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)
\end{aligned}$$

where the kinematic viscosity

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

is introduced.

The Navier-Stokes equations written in vector notation are:

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

2.2 Boundary equations

2.2.1 Solid walls

The Navier-Stokes equations have to be completed with boundary conditions. Usually, for viscous fluids, these will be: $\mathbf{u} = \mathbf{0}$ on the boundary. This describes two phenomena: impermeability of the solid wall (no fluid can flow through the wall) and the fact that the fluid sticks to the wall due to its viscosity. The latter is often called the **no-slip** condition.

2.2.2 Free surface

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

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

In this case :

- $u_n = \mathbf{u}\cdot\mathbf{n}$ denotes the velocity in the direction normal to the surface,
- $u_t = \mathbf{u}\cdot\mathbf{t}$ denotes the velocity in tangential direction,
- p_0 is the pressure of the air above the droplet,
- μ is the viscosity,

- γ is the surface tension,
- H represents the curvature of the surface.

It is also necessary to keep track of the free surface. Therefore we use an indicator function $\mathbf{F}(x,y)$ where $\mathbf{F}$=1 if there is fluid present and $\mathbf{F}$=0 elsewhere.

2.3 The axisymmetric case

SAVOF96 is capable of simulating fluid flow in axisymmetric containers as well. For axisymmetric calculations an extra dimension is required: **the azimuthal direction**. In this case cylindrical coordinates(r , φ , z) are used and w is introduced as the azimuthal velocity. Using this notation, the equation for the conservation of mass becomes:

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

The momentum equations for the radial, axial and azimuthal direction, respectively, become:

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

The mathematical equations are often written in dimensionless form.
In that case the following dimensionless groups are created:

- Reynolds number Re:

$$\frac{v_0 r_0}{\nu}$$

- Weber number We:

$$\frac{\rho {v_0}^2 r_0}{\gamma}$$

- Froude number Fr:

$$\frac{{v_0}^2}{r_0 g}$$

where v_0 is the preimpact droplet velocity, r_0 is the preimpact droplet radius, ρ is the density, γ is the surface tension coëfficient, ν is the kinematic viscosity of the liquid and g the gravitational acceleration.

Chapter 3

Numerical model

3.1 The Poisson-equation for the pressure

In this section a Poisson equation is derived from the incompressible Navier-Stokes equations. SAVOF96 uses this Poisson equation to compute the pressure.

We use the following abbreviation :

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

($\mathbf{R}$ contains all of the convective, diffusive and external forces)

If the density ρ is normalized ($\rho = 1$) we can write the Navier-Stokes equations as:

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

The $\frac{\partial u}{\partial t}$ term is discretized in time with a forward Euler method yielding:

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

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

Here n+1 and n denote the new and old time level, respectively and δt denotes 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:

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

Substituting this in (3.1), we obtain:

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

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

3.2 Discretization

SAVOF96 uses a Cartesian grid for its discretization with optional refinements near user-specified locations.

The unknown 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 edges of the cell, the vertical velocity v in the middle of the horizontal edges and the pressure p in the cell centres.

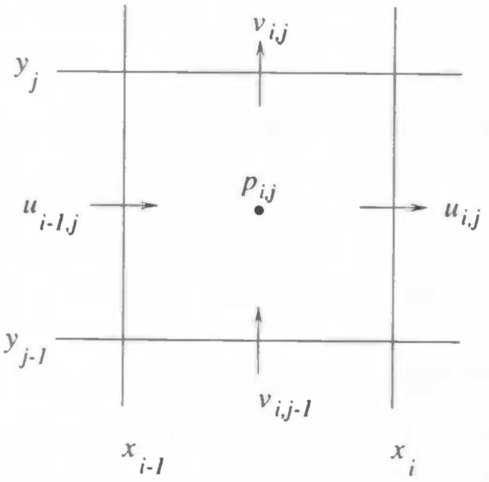


Figure 3.1: Marker-And-Cell method

The Poisson equation

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

is solved in a few steps.

First the momentum equation is integrated, i.e. $\frac{u^n}{\delta t} + \mathbf{R}^n$ is calculated.
We will show how this is done for the x-direction.

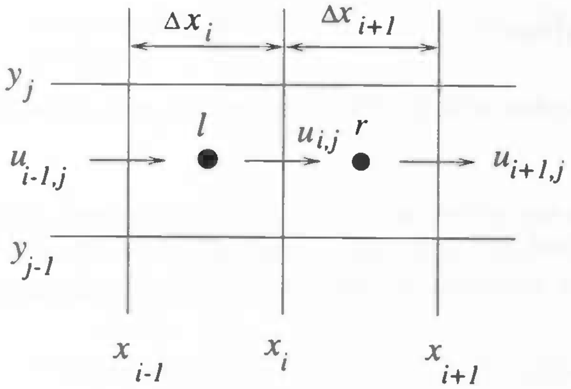
Consider the viscous terms in $\mathbf{R}^n$:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

The second order derivative is discretized centrally.
First:

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

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



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

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

In this manner the diffusive terms are calculated.

Now consider the convective terms:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}$$

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

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

($\alpha = 1$ 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 = div grad
- $x = p^{n+1}$
- b = div $(\frac{\mathbf{u}^n}{\delta t} + \mathbf{R}^n)$

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 a 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 U whose product equals A. Those ignored fill-in elements can result in unreliable solutions when thse 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:

- Take $\boldsymbol{x}^{(0)} = \boldsymbol{p}$;

  Calculate $\boldsymbol{r}^{(0)} = \boldsymbol{p} - A\boldsymbol{x}^{(0)}$

  and also $\boldsymbol{z}^{(0)} = K^{-1}\boldsymbol{r}^{(0)}$

  Compute for n=0,1,2,... the vectors $\boldsymbol{x}^{(n+1)}$, $\boldsymbol{r}^{(n+1)}$ and $\boldsymbol{z}^{(n+1)}$ from
- $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \alpha_n \boldsymbol{z}(n)$ with $\alpha_n = (\boldsymbol{x}^{(n)},\ \mathbf{K}^{-1}\boldsymbol{r}^{(n)}) \ / \ (\boldsymbol{z}^{(n)},\ A\boldsymbol{z}^{(n)})$
- $\boldsymbol{r}^{(n+1)} = \boldsymbol{r}^{(n)} - \alpha_n A\boldsymbol{z}^{(n)}$
- $\boldsymbol{z}^{(n+1)} = \mathbf{K}^{(-1)}\boldsymbol{r}^{(n+1)} + \beta_n \boldsymbol{z}^{(n)}$ with $\beta^n = (\boldsymbol{r}^{(n+1)}, \mathbf{K}^{(-1)}\boldsymbol{r}^{(n+1)}) \ / \ (\boldsymbol{z}^{(n)}, \mathbf{K}^{(-1)}\boldsymbol{z}^{(n)})$

This algorithm has been implemented in the subroutine MILU and gives us $p^{(n+1)}$

The new velocity can be calculated using:

$$u_{i,j}^{n+1} = (u_{i,j}^{n} + \delta t \mathbf{R}_{i,j}^{n}) - \delta t \text{ grad } p_{i,j}^{n+1}$$

where the gradient is discretized as

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

3.3.1 SOR

An alternative iteration method for vectorcomputers is the SOR (Successive OverRelaxation) method.

SAVOF96 uses a **Red-Black** or **Checkerboard ordening**, that is the pressure matrix is divided into 'red' and 'black' grid points. These points are stored into one-dimensional vectors which enhances vectorization.

Let D be a vector containing the diagonal elements of A, L the lower triangular matrix of -A and U the upper triangular matrix of -A (that is A=D-L-U). The SOR method with relaxation parameter ω is defined as

$$x^{k+1} = C_\omega x^k + Q^{-1}b$$

where

- $C_\omega = (D - \omega L)^{-1}((1-\omega)D + \omega U)$,
- $Q = \frac{1}{\omega}D - L$.

This equation is solved in subroutine SOLVEP, automatically adjusting ω for optimal convergence. This gives the pressure p^{n+1}. Subsequently the new velocity can be calculated using

$$u_{i,j}^{n+1} = (u_{i,j}^n + \delta t R_{i,j}^n) - \delta t \text{ grad } p_{i,j}^{n+1}$$

The gradient is discretized as

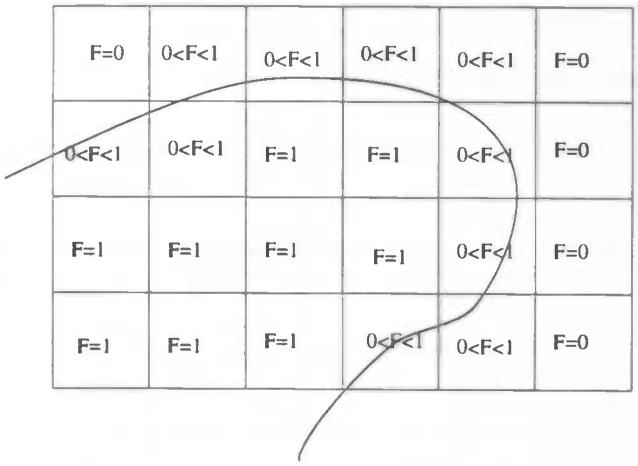
$$\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\frac{1}{2}\Delta x_i + \frac{1}{2}\Delta x_{i+1}}$$

This is also done in SOLVEP.
SOR is a fairly simple process; the performance of SOR for simple cases (that require only a few iterations) is very good. Another advantage of SOR is that it vectorizes and parallelizes excellently.

3.4 Description of a free surface

A means of keeping track of the free surface of a fluid during calculation is necessary to know where to apply the momentum equation. Evidently, this only has to be done at cells that contain fluid. In SAVOF96, use is made of an **indicatorfunction**. This is a function F(i,j) that is 1 in the case that cell(i,j) is completely filled with fluid and 0 if it is empty. Also values in between can be attained, depending on the percentage of the cell that is filled. Besides the indicator function, there is the cell labeling 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. It 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 to the bottom
- 4: surface cell with full cell to the top
- 5: degenerated cell
- 6: empty cell
- 7: outflow cell



F=0	0<F<1	0<F<1	0<F<1	0<F<1	F=0
0<F<1	0<F<1	F=1	F=1	0<F<1	F=0
F=1	F=1	F=1	F=1	0<F<1	F=0
F=1	F=1	F=1	0<F<1	0<F<1	F=0

- 8: inflow cell
- 9: obstacle or boundary cell

Using the above labeling, the position where the momentum equation must be applied can be identified: at cell faces between full, surface and/or outflow cells. At faces between surface and empty cells boundary conditions are applied, to be described in the next section.

3.5 Boundary conditions

As noted earlier, the boundary conditions for the Navier-Stokes equations usually are $\mathbf{u}$=0 at solid walls (no-slip condition). Since the velocities are not always defined at 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 (i=1). The horizontal velocity u is not defined on the wall, but half a grid point higher. So what we do is the following: set $u(0,j)$ = -$u(1,j)$ so that after interpolation of these two velocities the resulting velocity at the wall is zero.

At the free surface of a fluid, the governing boundary conditions are:

$$-p + 2\mu\frac{\partial u_n}{\partial n} = -p_0 + 2\gamma H \quad (3.3)$$

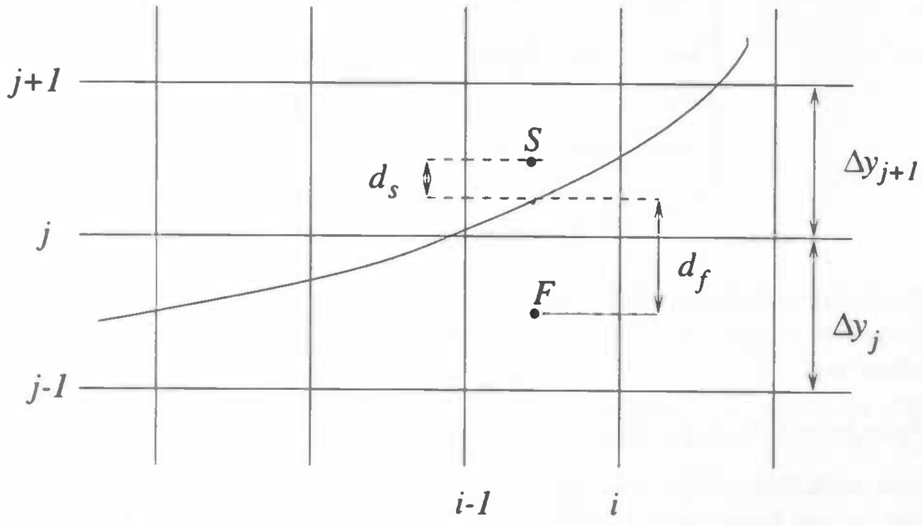
$$\mu\left(\frac{\partial u_n}{\partial t} + \frac{\partial u_t}{\partial n}\right) = 0 \quad (3.4)$$

In this case :

- $u_n = \mathbf{u}\cdot\mathbf{n}$ denotes the velocity in the direction normal to the surface,
- $u_t = \mathbf{u}\cdot \mathrm{t}$ denotes the velocity in tangential direction,
- p_0 is the pressure of the air above the droplet,
- μ is the viscosity,

- γ is the surface tension,
- H represents the curvature of the surface.

The curvature is calculated using information from the indicator function. The pressure at the free surface needed in (3.3) is interpolated as follows:



Suppose we have a full cell ($F_{i,j} = 1$, pressure p_F) with above it a cell that is only partially filled ($0 < F_{i,j+1} < 1$, pressure p_S). Then we can estimate the average height of the surface in the upper cell to be:

$$F_{i,j+1}\Delta y_{j+1}$$

so the distance from the center of the upper cell to the surface is

$$d_S := \frac{1}{2}\Delta y_{j+1} - F_{i,j+1}\Delta y_{j+1}$$

and the distance from the center of the lower cell to the surface is

$$d_F := \frac{1}{2}\Delta y_j - F_{i,j+1}\Delta y_{j+1}$$

Since the distance between the centers of the cells is

$$d := \frac{1}{2}\Delta y_j + \frac{1}{2}\Delta y_{j+1}$$

we can now linearly interpolate the pressures in the two cells to obtain the pressure at the free surface:

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

Because the momentum equations have to be calculated in between surface cells as well, we need to establish extra velocities on those cell faces. To overcome this problem, we apply the continuity equation div $\mathbf{u} = 0$ in surface cells as well. If we have three velocities available, then this procedure gives the fourth. If less velocities are available, we take e.g.

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0$$

The remaining velocities required, lying just outside the surface cell layer, are calculated using (3.4).

Chapter 4

Results

In order to illustrate the capability of the program SAVOF96 to produce physically acceptable solutions, a wide array of calculations were performed involving a variety of process parameters.
All calculations started with 42 gridpoints in x- and y-direction, resulting in 1764 points. After that a finer grid was used, up to 40.000 points. The computations for small grids were performed on a Hewlett Packard workstation. After grid refinement (e.g. 200*200) the calculations were done on a supercomputer: the Cray J932.

4.1 Deformation of a liquid droplet impinging upon a flat surface

4.1.1 Waterdroplets

The first simulation performed was chosen to be characteristic of typical spray cooling conditions.

CASE 1

A waterdroplet of radius $r_0 = 0.0345$ cm was considered to impinge on a stationary flat plate with velocity $v_0 = 146$ cm/sec.
For water, the following values were used :

- surface tension coëfficiënt $\gamma = 0.073$ N/m
- density $\rho = 1000$ kg/m^3
- kinematic viscosity $\nu = 10^{-6}$ m^2/sec

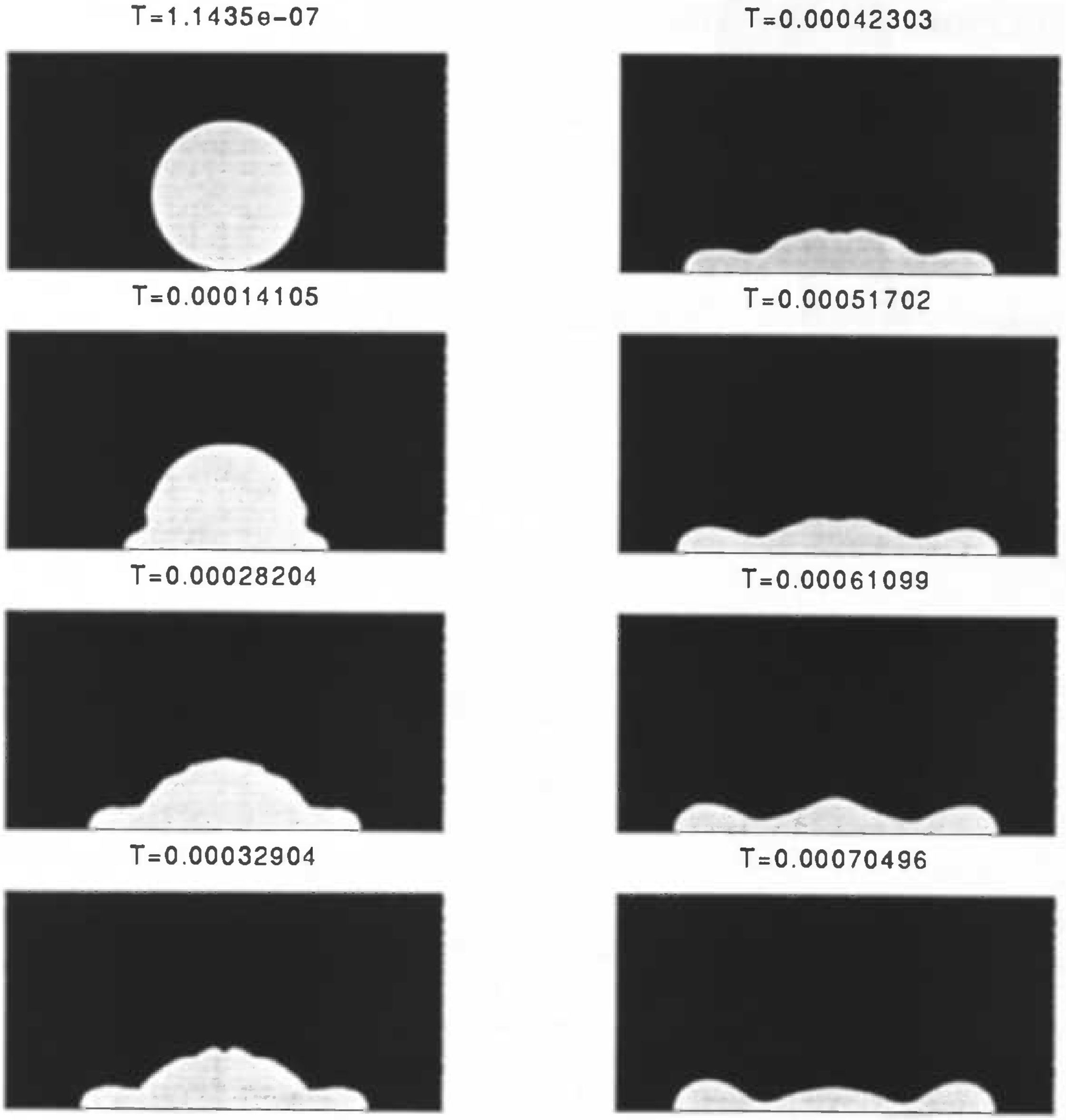


Figure 4.1: Splatting sequence of a water droplet with $r = 0.0345$ cm and v=146 cm/sec. on a 175*175 grid

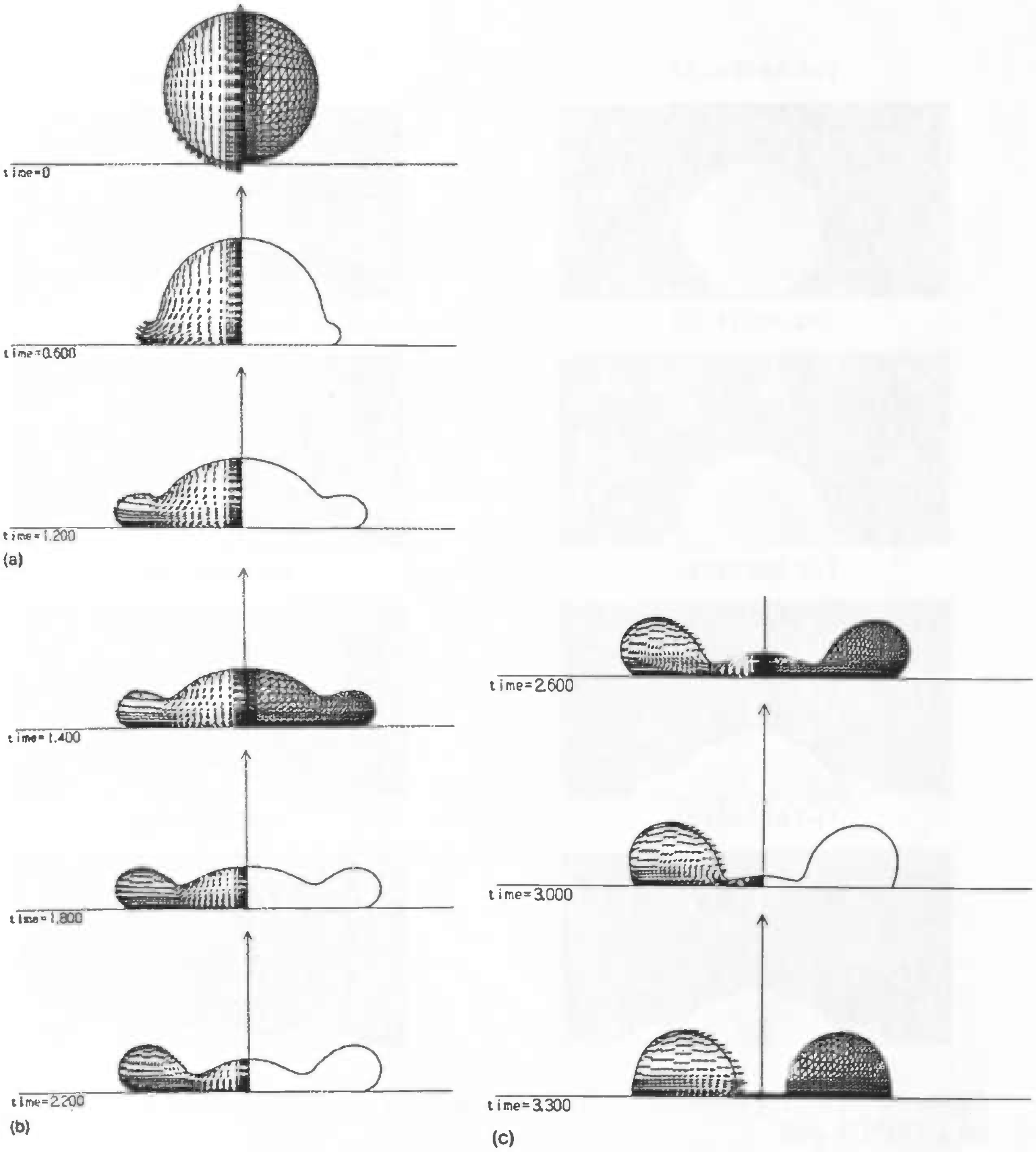


Figure 4.2: Results of CASE 1 according to Fukai[1]

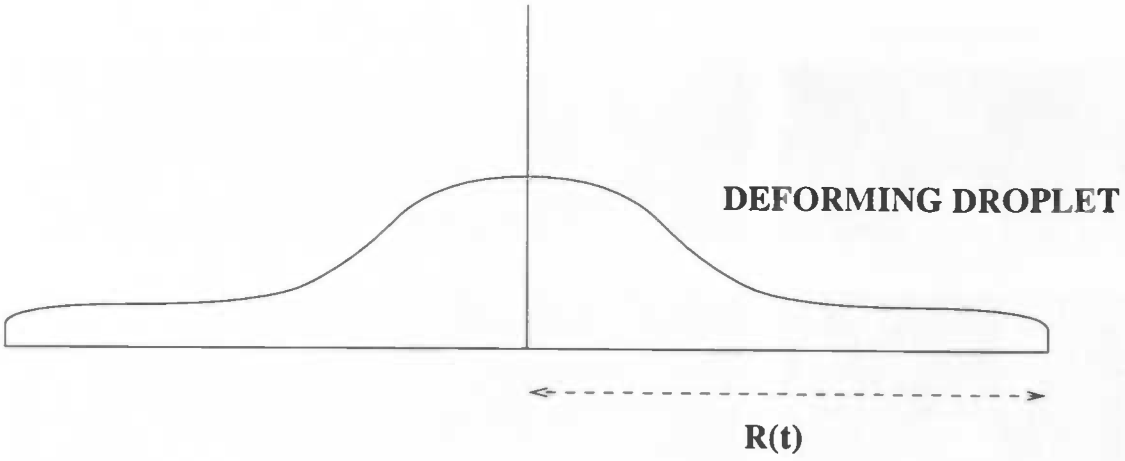


Figure 4.3: Definition of the maximum splat radius R(t) of a deforming droplet.

The above conditions resulted in the following values of the relevant dimensionless numbers :

- $Re \approx 500$
- $We \approx 10$
- $Fr \approx 630$

Fig.4.1 depicts a sequence of frames corresponding to different instances of the impact process. Immediately after contact, a ring structure is formed around the droplet which propagates rapidly in the radial direction and away from the axis of symmetry. The surface tension effects are primarily experienced by the laterally propagating flow which spreads over the flat plate. Surface tension effects tend to decelerate the droplet spreading and eventually dominate the spreading dynamics.

The computations were compared to those of Fukai[1], visualized in Fig.4.2 and a few differences were noticed. The pictures show that he used another time-scale, namely $\mathbf{time} = \frac{t}{r_0/v_0}$. The calculations with SAVOF96 show that the droplet stretches slower than in the case of Fukai[1]. In the last case the impact of the droplet viscosity seems less strong, compared to the results achieved with SAVOF96. For comparison, watch the last frame in Fig.4.1 at t=0.00070496 seconds and the frame in Fig.4.2 at **time**=3.0 (corresponding to t=0.00070808 seconds). Fukai[1] used a mathematical model based on a finite element method (while SAVOF96 is based on a finite difference method). His model also contained artificial compressibility. These facts can be the reason for the small differences.

The maximum splat radius R variation with time also describes the droplet deformation during impact. (The maximum splat radius R(t) is defined as the radial distance of the outermost point of the splash from the splat axis of symmetry at the moment t).

Fig.4.4 shows the time-dependance of the splat radius R(t) for CASE 1.

The pressure p on top of the droplet was also calculated. In Fig.4.5 you can see how the pressure varies with the time for the case described above.

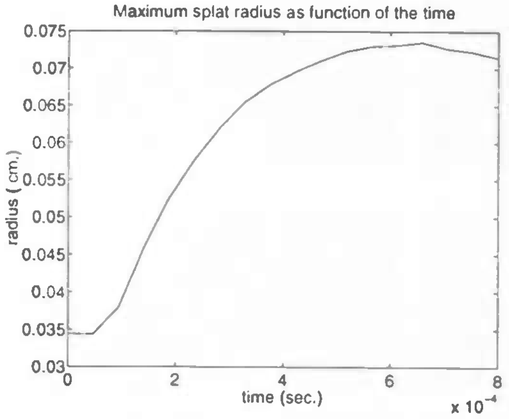


Figure 4.4: R_{max} for CASE 1

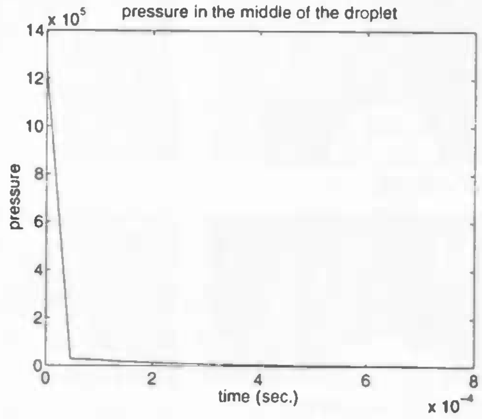


Figure 4.5: pressure for CASE 1

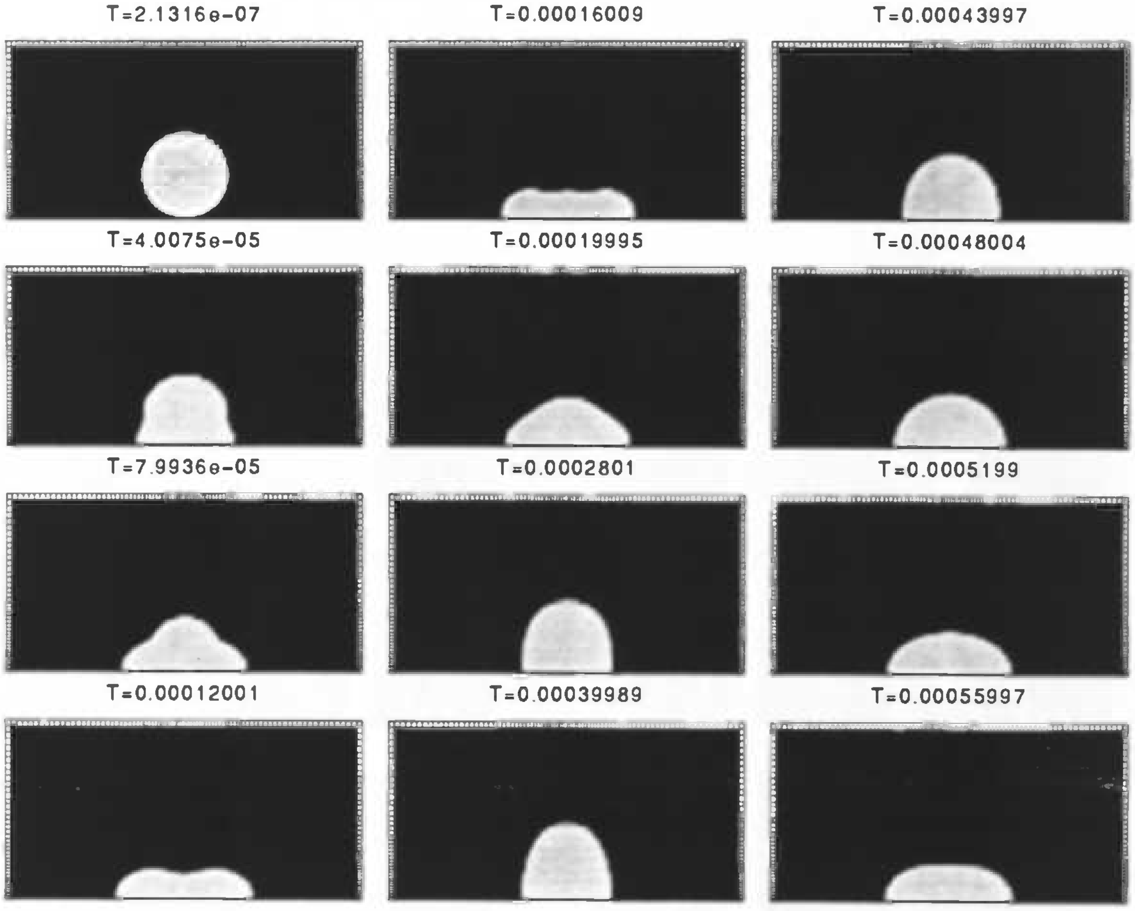


Figure 4.6: CASE 2 calculated on a 42*42 grid

CASE 2

A second simulation was performed for a water droplet of radius $r_0 = 0.01$ cm impinging on a stationary flat plate with velocity $v_0 = 100$ cm/sec .
These conditions resulted in the following values of the relevant dimensionless numbers :

- $Re \approx 100$
- $We \approx 1.4$
- $Fr \approx 1020$

We can compare the two figures 4.6 and 4.7 in order to observe the effect of grid refinement. Suprisingly, in this case the differences are not shockingly big. Exept for frame 4 (t=0.00012001) it looks very much alike.

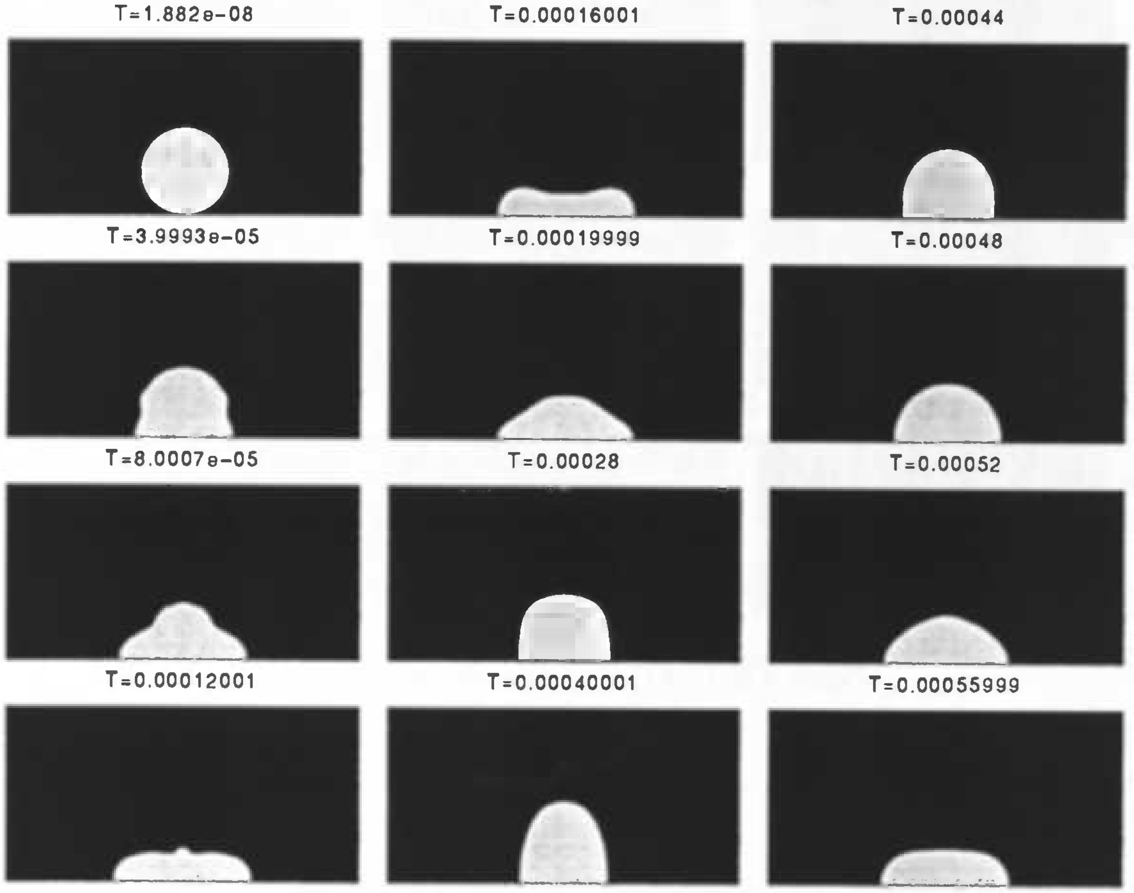


Figure 4.7: Splatting sequence of a water droplet with r=0.01 cm and v=100 cm/sec. on a 200*200 grid

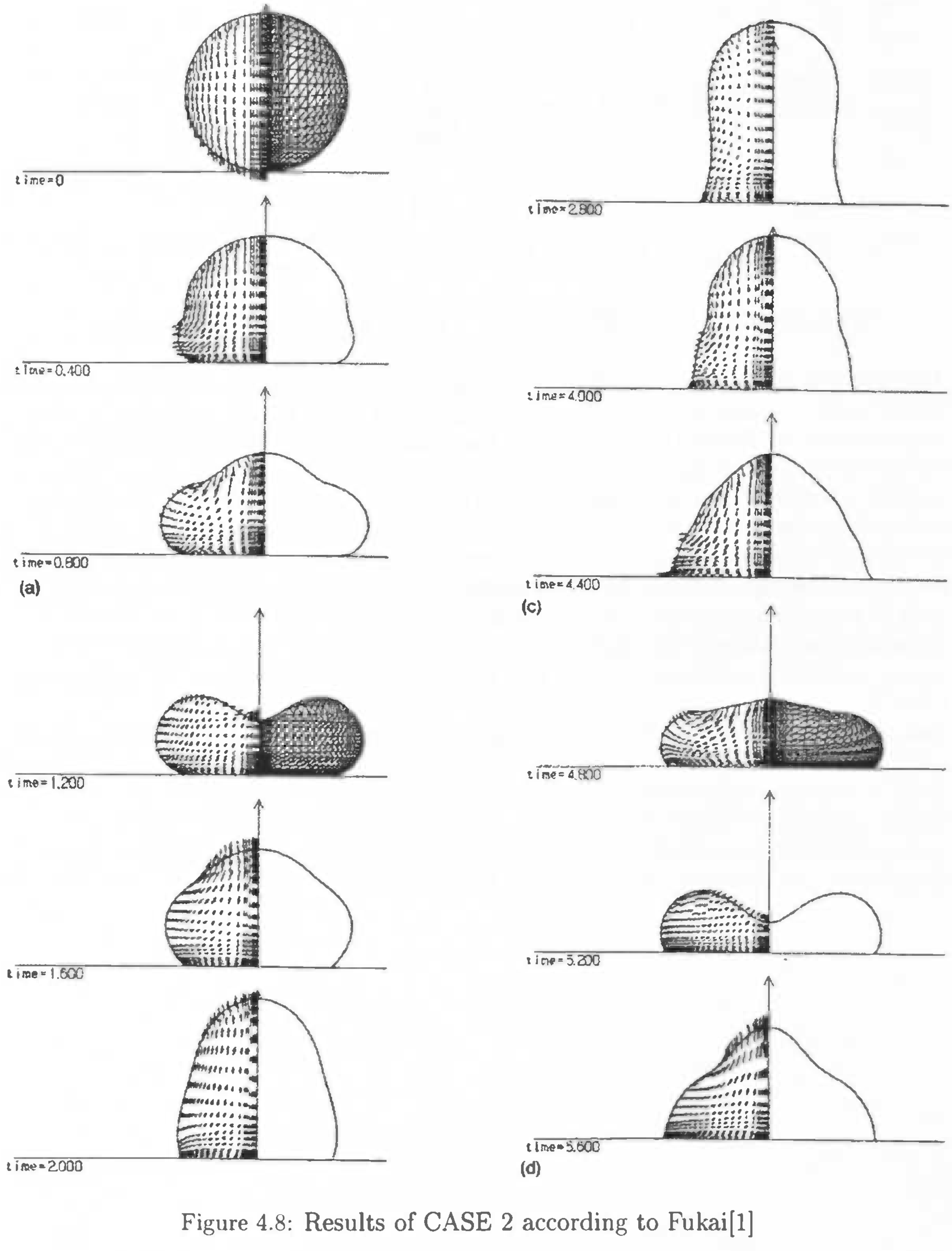


Figure 4.8: Results of CASE 2 according to Fukai[1]

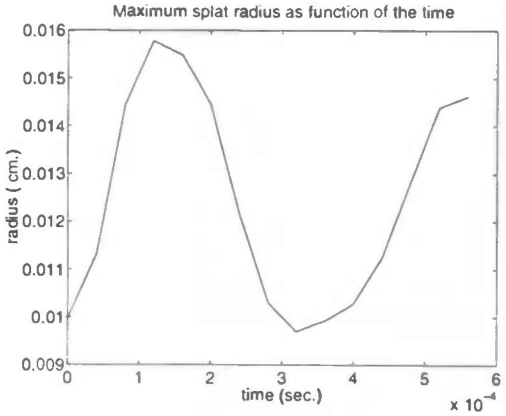


Figure 4.9: R_{max} for CASE 2

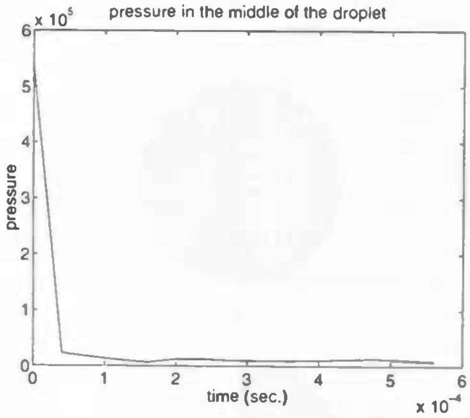


Figure 4.10: pressure for CASE 2

The low value of the Weber number associated with this run points out the very significant role of surface tension in the droplet spreading process. Fig.4.6 depicts a sequence of frames corresponding to different instances of the impaction process. It is apparent that the strong surface forces impedes the lateral downward motion and stall the liquid spreading at an early stage (t = 0.00016 seconds) when the splat thickness is still very large. The subsequent recoiling of the droplet causes an upward bulk motion that results in the elongation of the splat in the axial direction. This vertical movement is counteracted by the gravity forces which eventually dominate and reverse the direction of the flow (t = 0.00044 seconds). A second cycle of the same events is thus established.
For comparison, observe the results of Fukai[1] in Fig.4.8.
Again, SAVOF96 produces a droplet that moves slower, so more viscous than in the case of Fukai[1].
Fig.4.9 and Fig.4.10 show the time-dependance of the splat radius $R(t)$ and the pressure for CASE 2.

4.1.2 Metal droplets

CASE 3

An additional simulation was conducted involving a liquid-tin droplet of radius $12 * 10^{-4}$ cm impinging on a stationary flat plate with velocity $v_0 = 2500$ cm/sec. These conditions were chosen as typical for spray coating applications.

For liquid tin the following values were used :

- surface tension coefficient $\gamma = 0.554$ N/m
- density $\rho = 7000$ kg/m^3
- kinematic viscosity $\nu = 2.6 * 10^{-7}$ m^2 /sec.

The above conditions resulted in the following values of the relevant dimensionless numbers :

- $Re \approx 1200$
- $We \approx 100$
- $Fr \approx 5.6 * 10^4$

Fig.4.11 depicts a sequence of frames corresponding to different instances of the metal droplet impaction process. The notable difference between the metal droplet Fig.4.11 and the water droplet spreading sequences Fig.4.1 and Fig.4.6 is a result of the substantially different values of the relevant dimensionless flow numbers.

Fig.4.11 shows that immediate after contact, a thin film is formed at the periphery of the splat. In this case it appears that the impeding effects of surface tension to the overall spreading process take longer to dominate. This trend is expected, due to the higher value of We for the tin droplet compared to the water droplet (100 vs. 10 or 1.4). As seen in Fig.4.11, the droplet stretches to a significant degree before spreading is halted by the dominance of the surface tension mechanisms (t > 2.8003e-06 seconds). In Fig.4.12 the results of Fukai[1] have been listed.

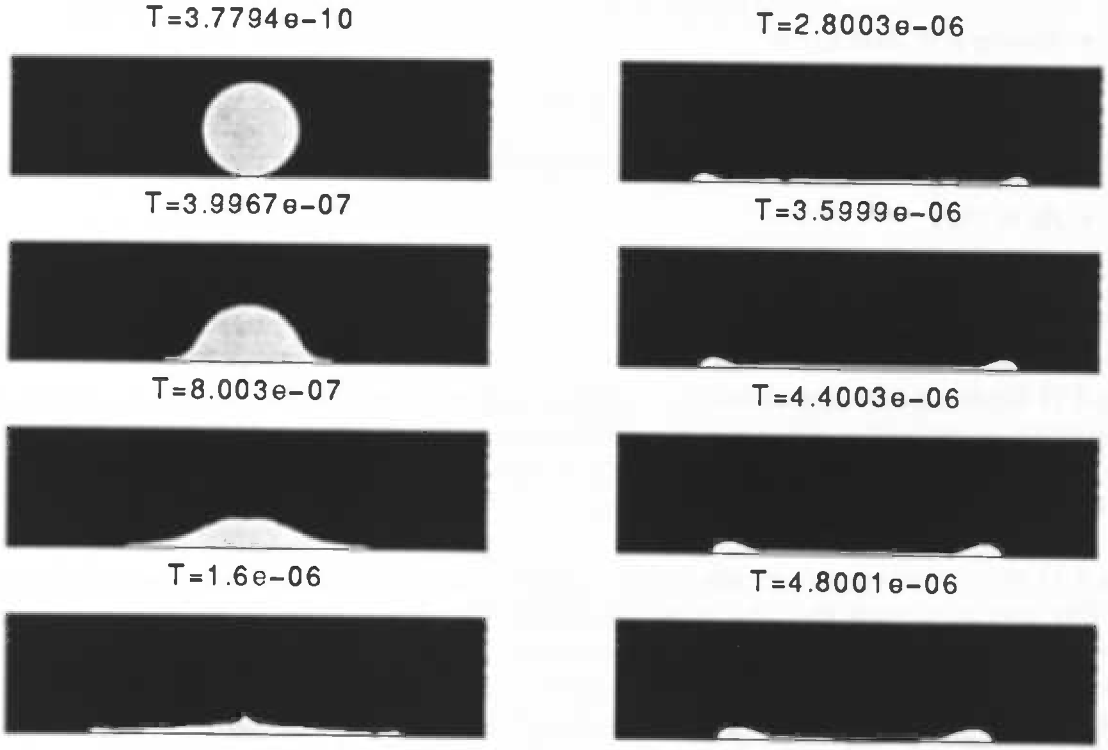


Figure 4.11: Splatting sequence of a liquid-tin droplet with $r = 0.0012$ cm and v=2500 cm/sec. on a 100*100 grid

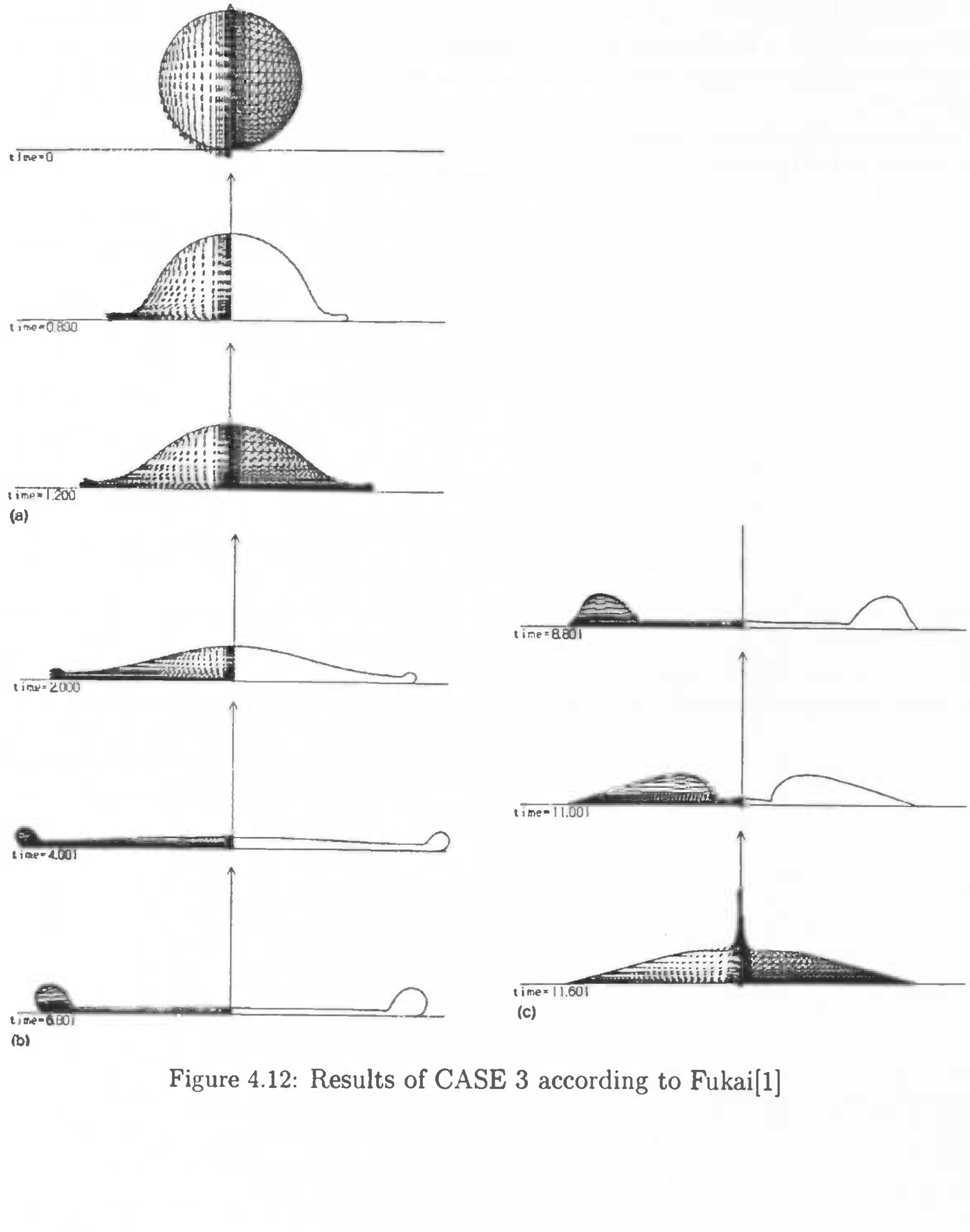


Figure 4.12: Results of CASE 3 according to Fukai[1]

Summary:

In the three cases described above, the maximum splat thickness did not always occur at the axis of symmetry. At late times it usually occurred at the periphery of the splat, where mass accumulation was observed.

The periphery of the splat initially advanced outward. This advancement was retarded and finally halted by the action of surface tension and viscosity. Subsequently, flow reversal was observed and the main flow was directed toward the axis of symmetry.

Significant differences were observed between the flow fields of a water droplet and a liquid tin-droplet.

Furthermore, the droplet movement calculated by SAVOF96 seems slower compared to the results of Fukai[1], probably due to a higher numerical viscosity.

4.2 Wetting effects on the spreading of a liquid droplet colliding with a flat surface

In this simulation special attention is paid to the contact-angle.
Two cases are distinguished namely :

- the contact-angle for the spreading stage : ψ_{CS}
- the contact-angle for the recoiling stage : ψ_{CR}

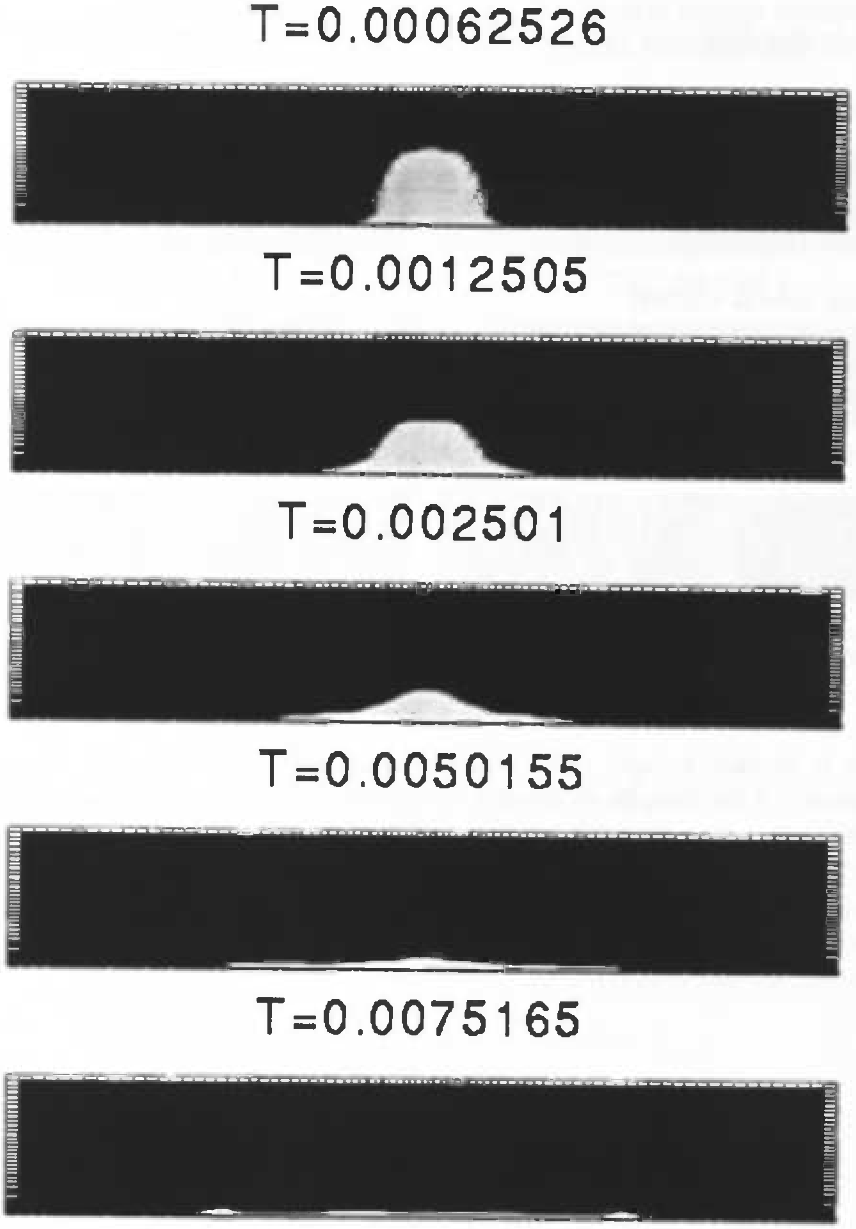
CASE 4

The following case is studied:
A water droplet with:

- radius $r_0 = 0.188$ cm
- velocity $v_0 = 150$ cm/sec.
- $\psi_{CS} = 60°$
- $Re \approx 3010$
- $We \approx 58.4$

The results are listed in Fig.4.13.

A thin film is formed around the droplet immediately after impact. Mass accumulation at the periphery of the droplet subsequently occurs and a ring structure is initiated at about t = 0.0075165 seconds. It shows that the ring structure develops at the periphery of the droplet as the fluid propagates outward in the radial direction. The results were compared to those of Fukai [2] in Fig.4.14 and show little difference.
In his article the following time-scale was used: $\tau = v_0 t / r_0$.
Fig.4.14 also shows the velocity-field.



T=0.00062526

T=0.0012505

T=0.002501

T=0.0050155

T=0.0075165

Figure 4.13: Splatting sequence of a water droplet with r = 0.188 cm and v = 150 cm/sec. (spreading) on a 100*100 grid

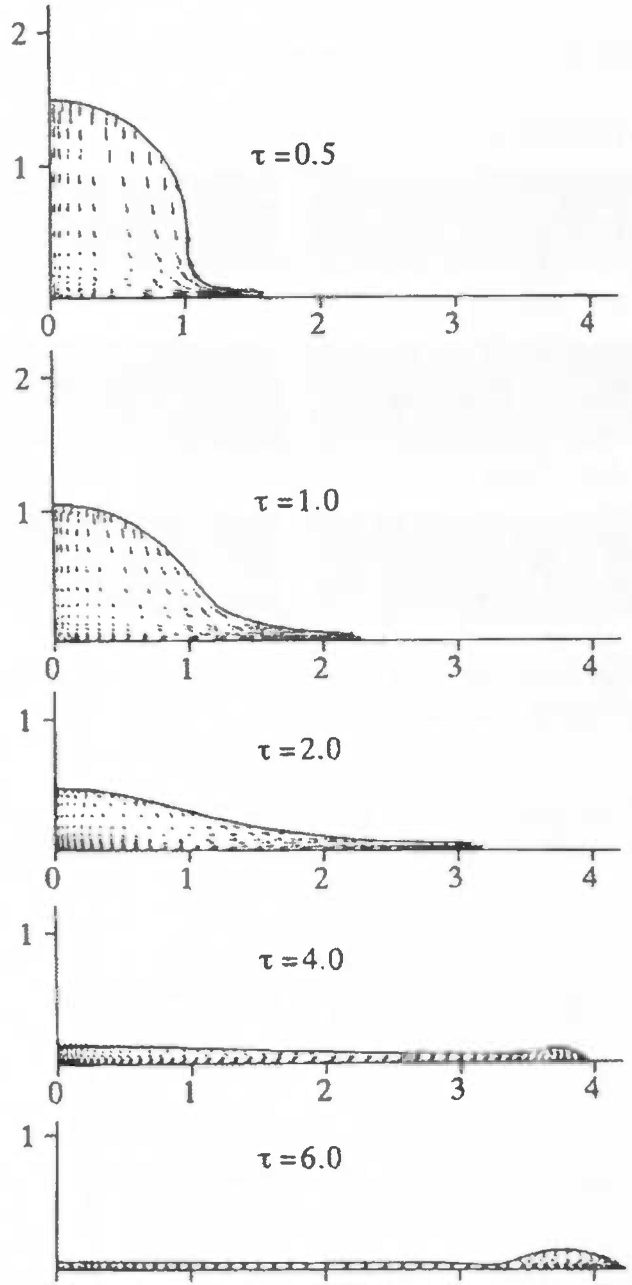


Figure 4.14: Spreading results according to Fukai[2]

CASE 5

Recoiling:

The frames in Fig.4.15 show the recoiling process. In this case the contact-angle $\psi_{CR} = 22°$.

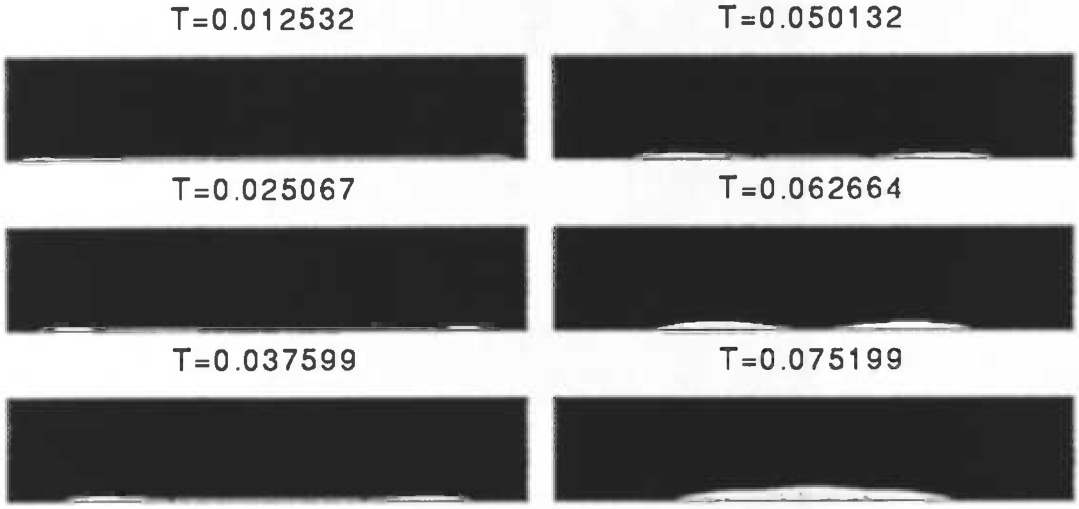


Figure 4.15: Splatting sequence of a water droplet with $r = 0.188$ cm and $v = 150$ cm/sec. on a 100*100 grid (recoiling)

As the spreading of the ring structure approaches the axis of symmetry, an upward bulk motion is observed (t=0.075197 seconds). Again, the results look pretty much the same as those of Fukai[2].

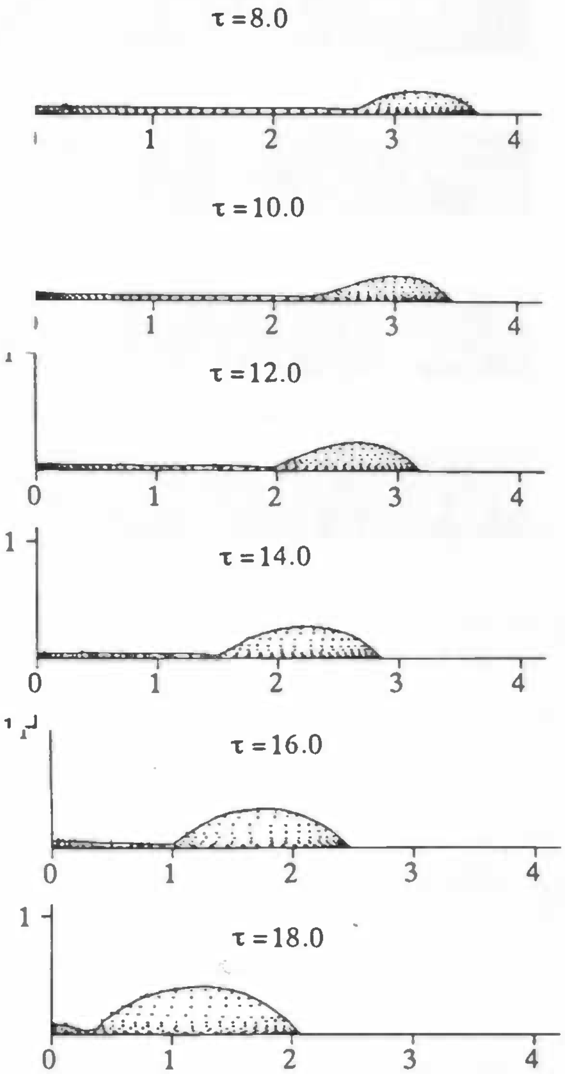


Figure 4.16: Recoiling according to Fukai[2]

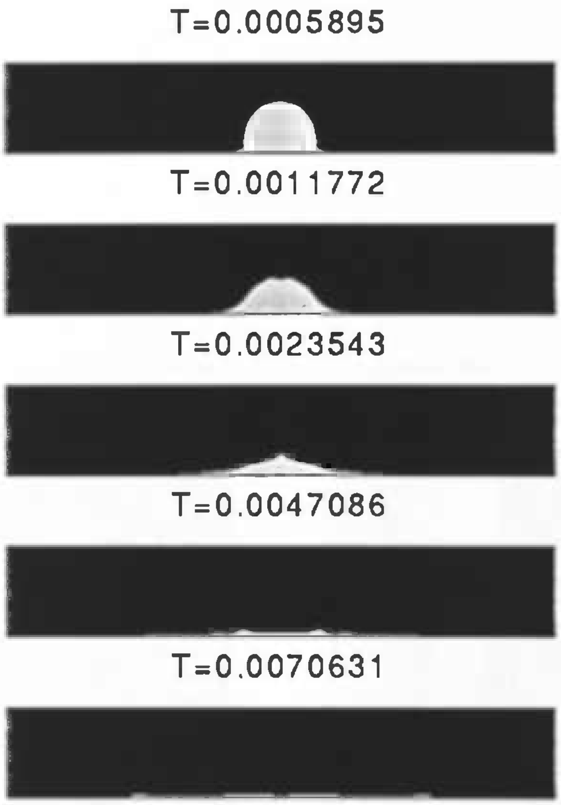


Figure 4.17: Splatting sequence of a water droplet with r = 0.186 cm and v = 158 cm/sec. (spreading) on a 100*100 grid

CASE 6

An additional simulation was performed with :

- $r_0 = 0.186$ cm
- $v_0 = 158$ cm/sec.
- $Re \approx 3130$
- $We \approx 64.1$
- $\psi_{CS} = 92^\circ$
- $\psi_{CR} = 60^\circ$

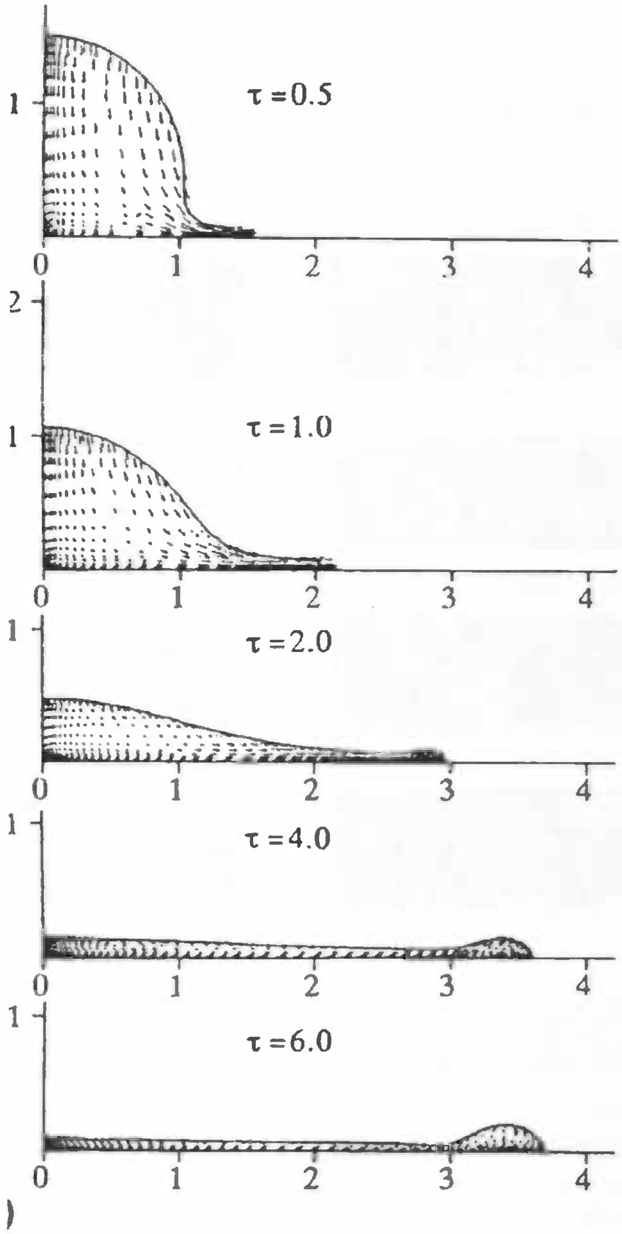


Figure 4.18: Spreading sequence according to Fukai[2]

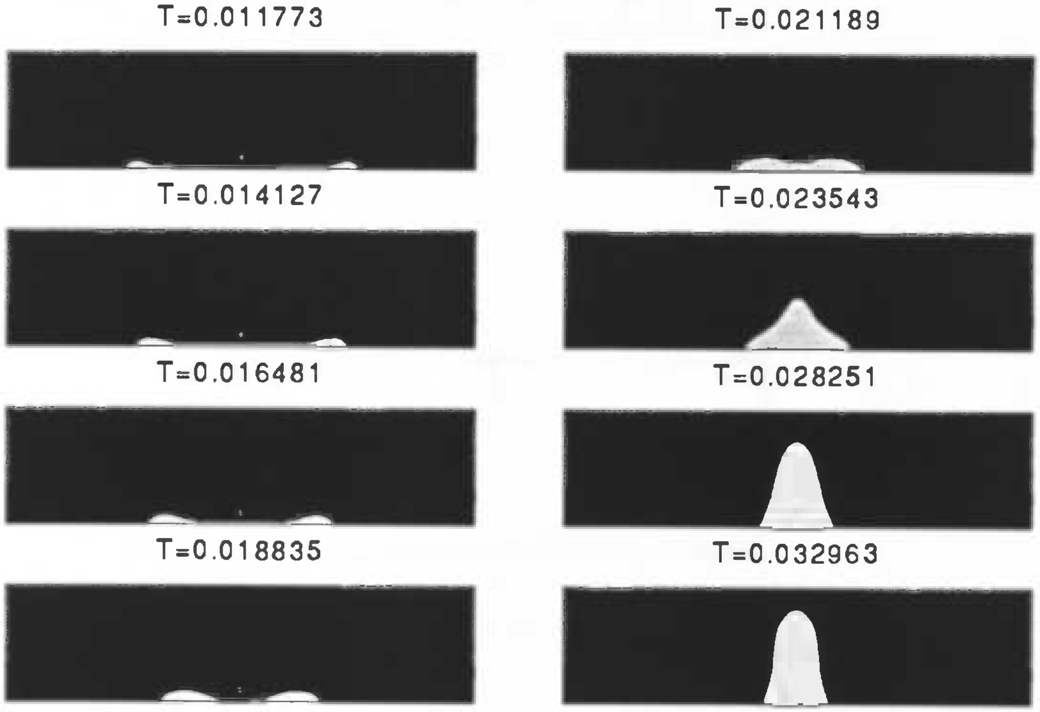


Figure 4.19: Splatting sequence of a water droplet with r = 0.186 cm and v = 158 cm/sec. on a 100*100 grid (recoiling)

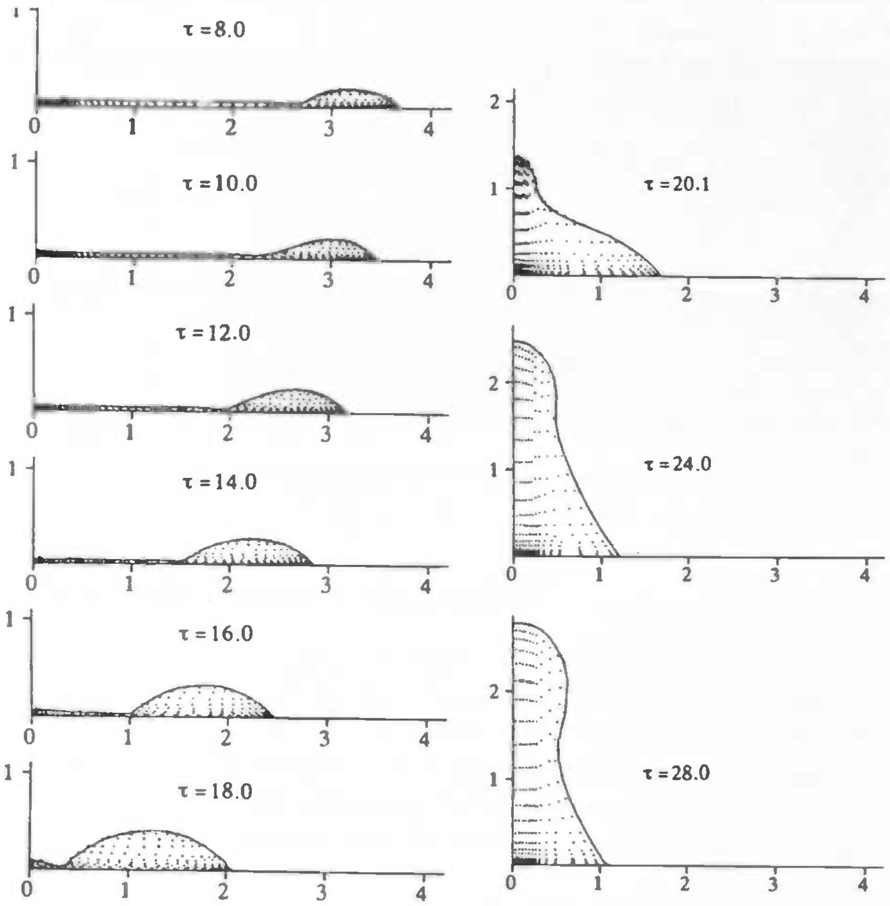


Figure 4.20: Recoiling sequence according to Fukai[2]

The results are shown in Fig.4.17 (spreading) and Fig.4.19 (recoiling).

They exhibit similarities as well as differences: the formation of a thin film, the subsequent development of a ring structure during both spreading and recoiling and the upward bulk motion. However, from the comparison of the impaction sequences it is apparent that the larger contact-angle corresponding to limited wetting in Fig.4.19 leads to a greater ring structure and strong upward bulk motion. This finding is reasoned as follows: The larger contact-angle causes a larger mean curvature at the periphery of the droplet. The surface tension force corresponding to this larger mean curvature decelerates the outward spreading process. Hence, more liquid mass is accumulated in the ring at the splat periphery. The restraining influence of the surface tension is indicated by the lower values of the splat radius of Fig.4.19 compared to Fig.4.17.

In the recoiling process, the large surface tension force mentioned above accelerates the inward flow, thus amplifying the upward bulk motion.

In Fig.4.18 and 4.20 the results of Fukai[2] have been depicted.

The first part - the spreading stage - shows little difference. The next part shows some deviation probably due to the fact that the calculations with SAVOF96 were done with $\psi_{CS} = 92°$.

From the pictures it is clear that this is correct in the spreading stage but not in the recoiling stage where $\psi_{CR} = 60°$ should have been used.
So the fact that both spreading and recoiling occur and SAVOF96 uses only one value for the contact-angle causes the difference.

Fukai[2] reported slightly different results for the cases 4-6.
The reason for these differences can be explained as follows:

1. He used triangular elements for the grid.
2. For the numerical time-integration he used an implicit method.
3. The two reasons mentioned before on page 22: using artificial compressibility and a finite element method.

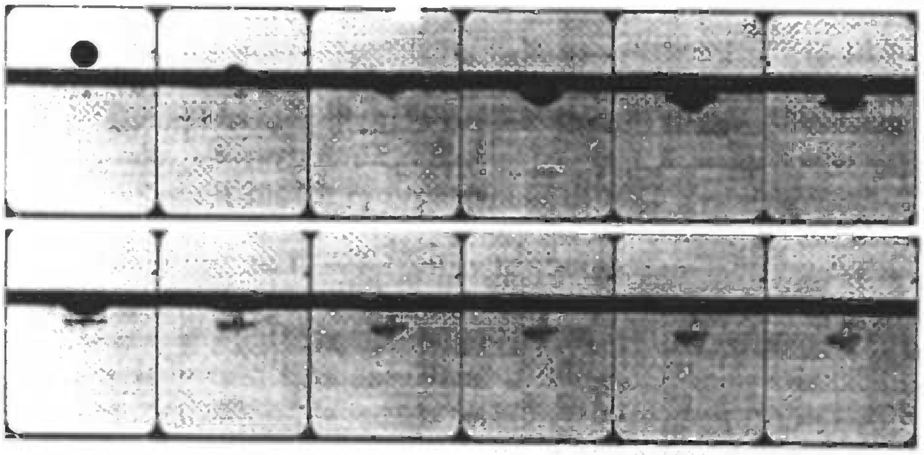


Figure 4.21: Photographs of CASE 7 according to Rein

4.3 A drop falling into a deep pool

When a drop of water enters a pool of water two different phenomena may be observed depending upon the diameter and velocity of the entering drops. In one case the drop causes a splash and in the other case the drop coalesces with the pool without a splash and as the drop penetrates the surface it forms a vortex ring. In this chapter also attention is paid to the transition from coalescence to splashing. In that case a jet is formed.
In the following calculations water is used for both the drop and the receiving liquid.

4.3.1 The formation of a weak vortex ring

Coalescence is usually connected with the formation of a vortex ring that propagates into the receiving liquid. Immediately after impact a crater is formed and the drop liquid spreads over is floor. The crater eventually recedes and a vortex ring appears. The surface of the receiving liquid assumes its horizontal equilibrium position and is not otherwise disturbed.

CASE 7

We consider the impact of a water drop on a deep liquid with impact conditions:

- radius $r = 0.124$ cm
- velocity $v = 87$ cm/sec.
- We ≈ 26
- Fr ≈ 31

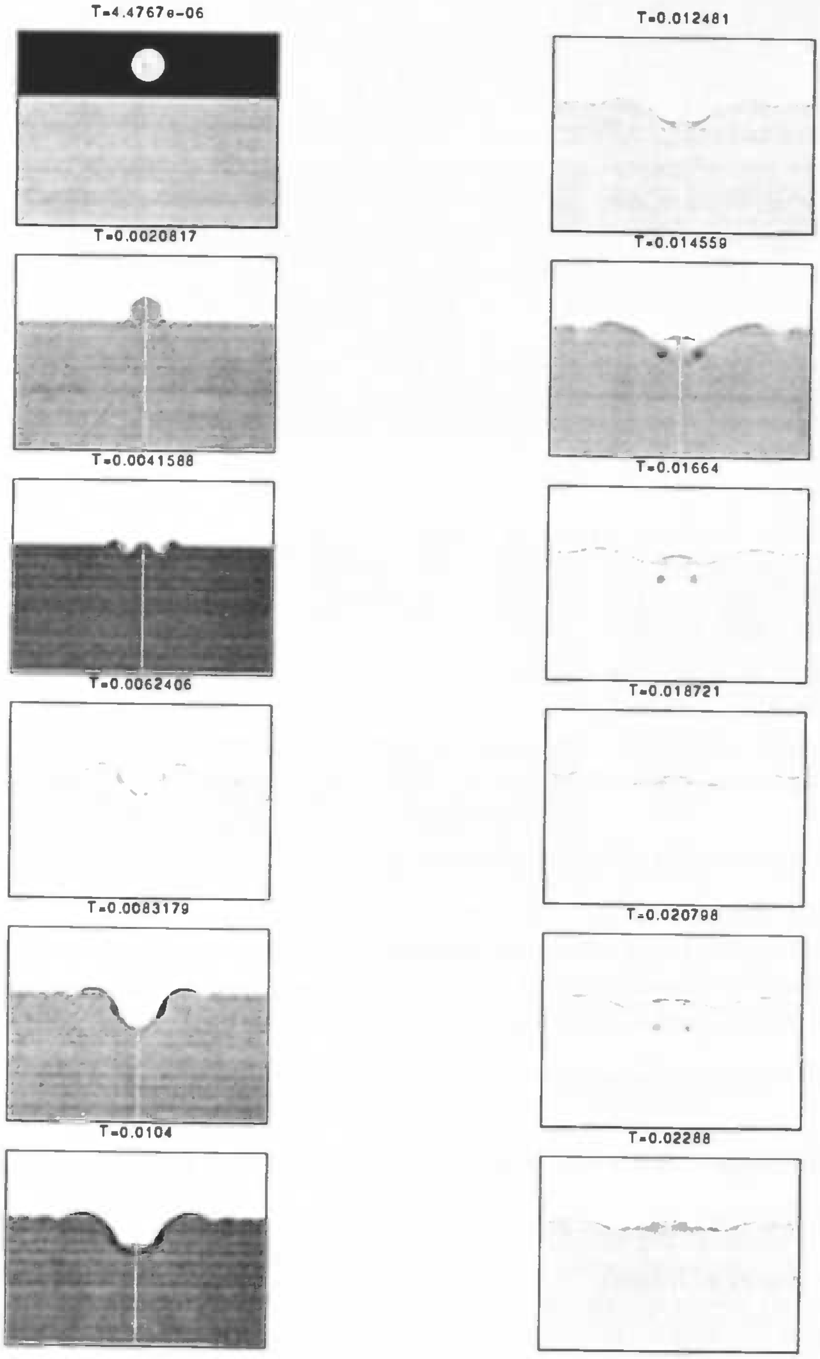


Figure 4.22: Splashing sequence of a water droplet with r = 0.124 cm and v = 87 cm/sec. on a 200*200 grid

The propagation velocity of the vortex ring was very small and it came quickly to a rest as shown in Fig.4.22. A crater, represented by a black region, can first be detected in frame 4. Drop liquid surrounds the crater, most of it being disturbed beside and little beneath the crater. The drop liquid starts to roll up into a vortex ring . When the crater recedes the vortex ring separates from the crater and begins to move slowly into the receiving liquid.
In Fig.4.22 plots of the vorticity of the droplet can be seen. In this simulation we choose for the vorticity (instead of velocity) because of the fact that the deforming process is not clearly visible when using velocity. The different grey colours correspond to different values of the vorticity.
For comparison, the results of Rein[3] have also been listed. He made photographs of the experiment, so in this case our calculations are compared with photographs (instead of calculations as in the former cases).

4.3.2 Formation of a central jet

We now consider a case in which a central jet is formed.

CASE 8

This case is presented in Fig.4.24 which shows a drop impact with

- radius $r = 0.115$ cm
- velocity $v = 223$ cm/sec.
- We ≈ 161
- Fr ≈ 220

After the crater has reached its maximum depth fluid begins to flow radially inward from the sides. After that the bottom of the crater begins to rise and a central jet is formed.
This experiment was conducteded by Rein[3] and his photographs can be seen in Fig.4.23.

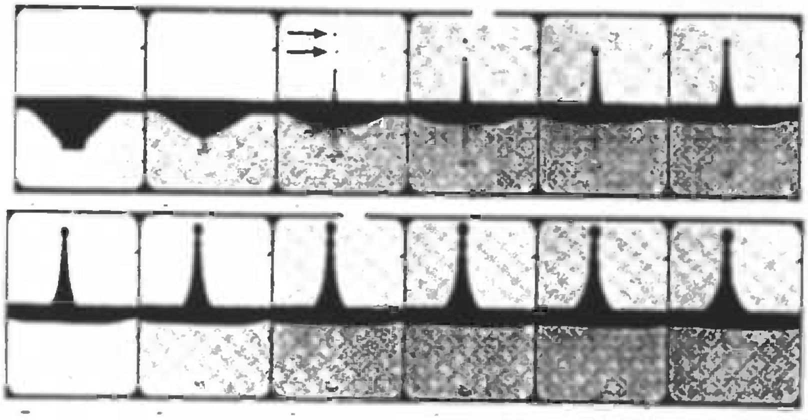


Figure 4.23: Results of CASE 8 according to Rein

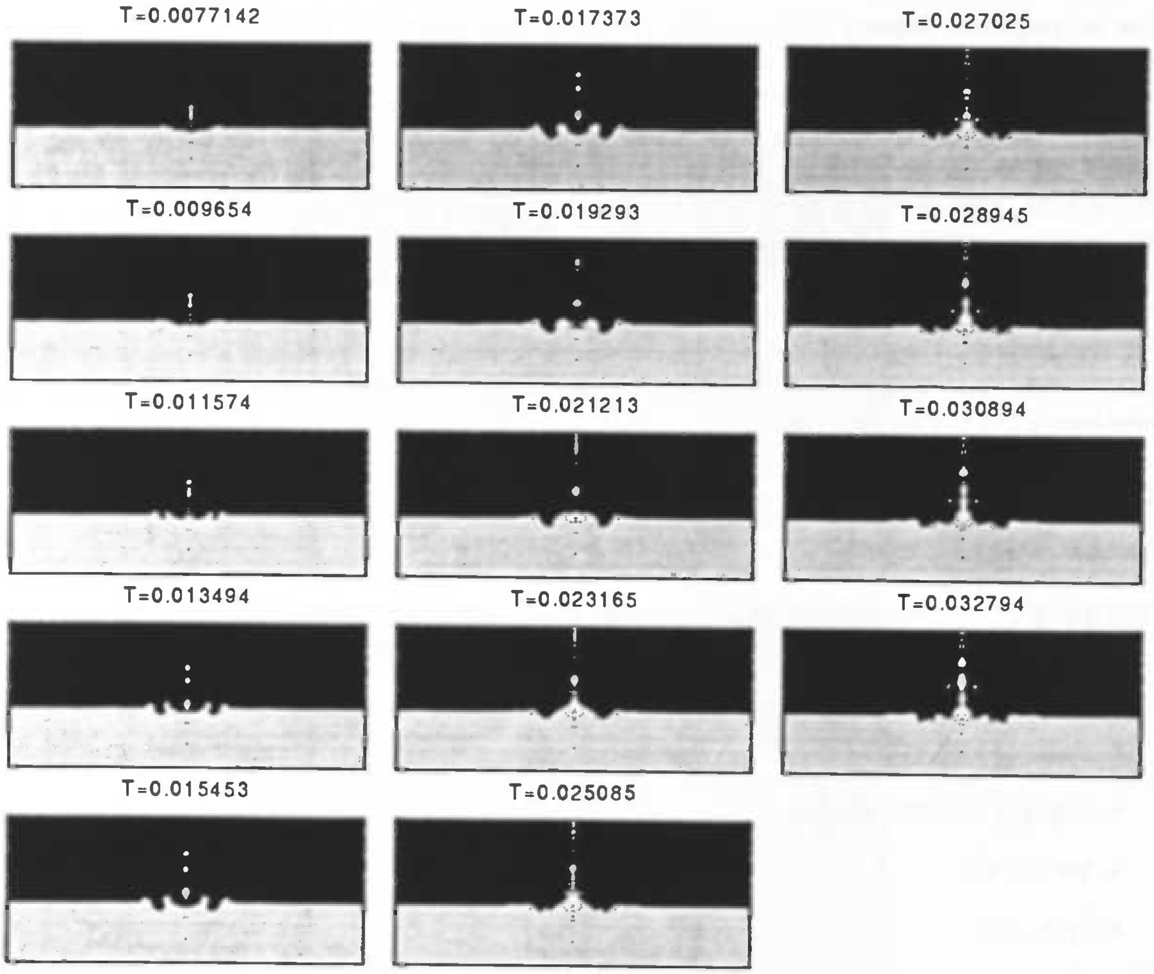


Figure 4.24: Splashing sequence of a water droplet with $r = 0.115$ cm and $v = 223$ cm/sec. on a 100*100 grid

The simulation of this jet showed quite different results compared to Rein[3] (not as smooth as the photographs but more "wild" and "fuzzy").

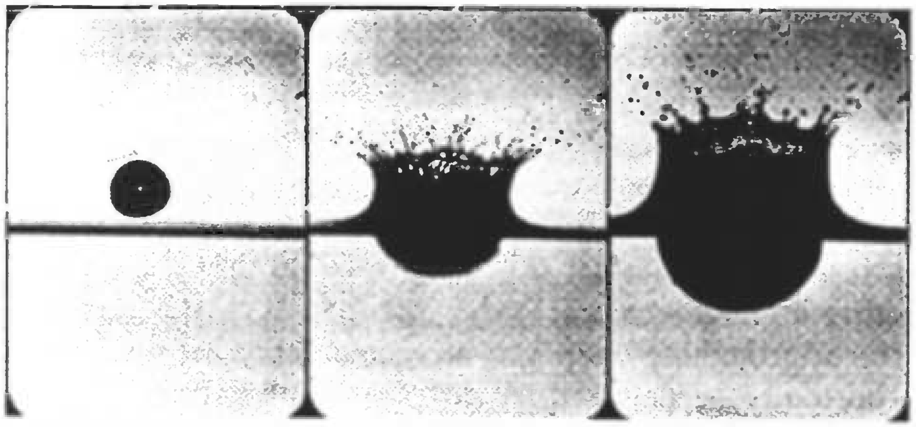


Figure 4.25: Results of CASE 9 according to Rein[3]

4.3.3 Splashing droplets

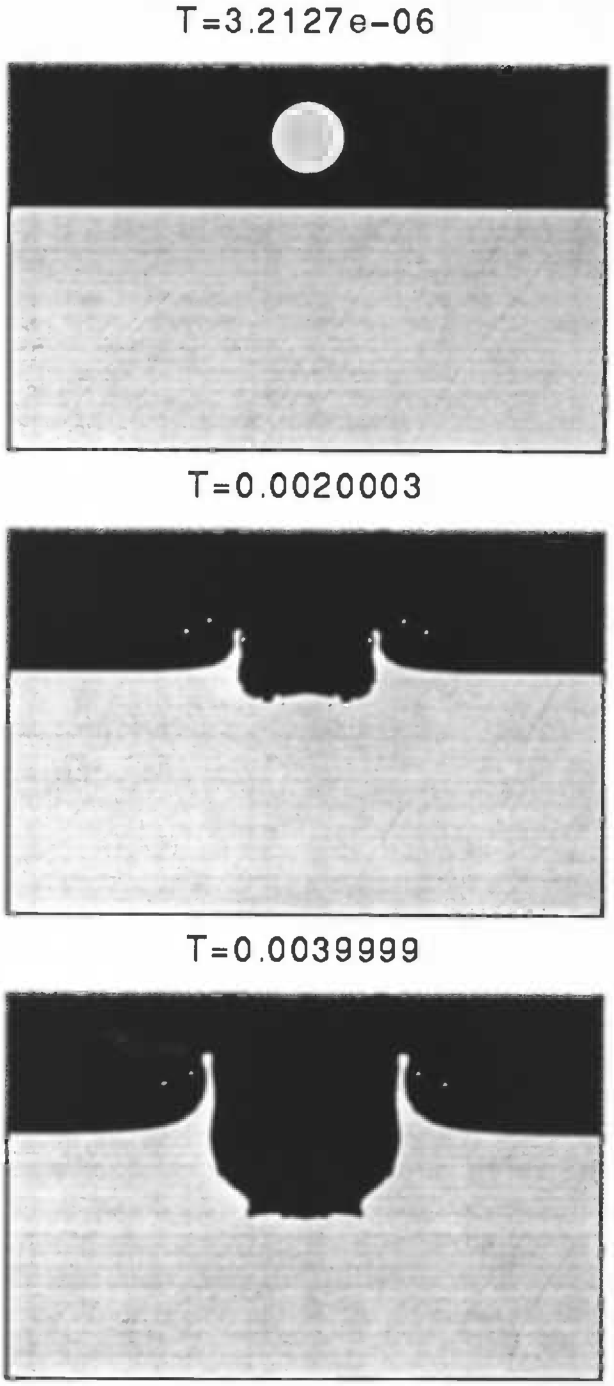
A splash includes not only a jet but also a liquid film rising from the periphery of the crater. The film breaks into filaments and the filaments break into drops to produce the corona form.

CASE 9

The following calculation was performed:

- radius $r = 0.25$ cm
- velocity $v = 500$ cm/sec.
- We ≈ 1750
- Fr ≈ 500.

SAVOF96 produced less outspoken results than in Fig.4.25 but it seems to resemble.
The same calculation done for a 42*42 grid produces less accurate results as can be seen in Fig.4.27. The deformation is obviously less sharp compared to the calculation on a 150*150 or a 200*200 grid. So in this case grid refinement does leed to better results as can be seen in the above example.



T=3.2127e-06

T=0.0020003

T=0.0039999

Figure 4.26: Splashing sequence of a water droplet with $r = 0.25$ cm and $v = 500$ cm/sec. (grid 200*200)

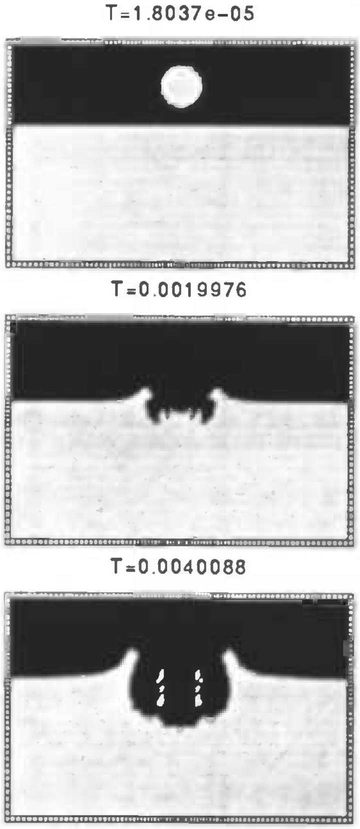


Figure 4.27: CASE 9 calculated on a 42*42 grid

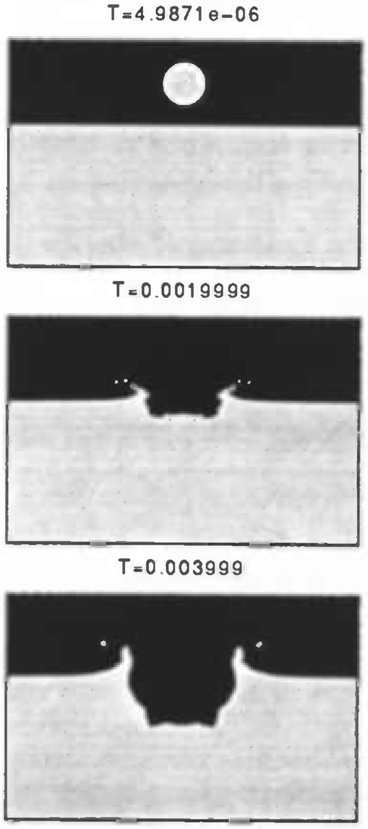


Figure 4.28: CASE 9 calculated on a 150*150 grid

Chapter 5

Conclusions

In this paper the deformation of **splatting** (impinging on a solid substrate) and **splashing** (impacting on a liquid surface) droplets has been studied. In the case of splashing droplets the following three phenomena have been observed namely:

- the formation of a vortex ring
- the formation of a central jet
- the formation of a crown

The program SAVOF96 was used for the numerical study and produced results in the form of velocity-, pressure-, streamline-plots etc.

Splatting droplets:
Several cases were tested and compared to the results of Fukai[1] and [2] who did the same experiments under the same conditions. All of them were handled very well by SAVOF96 and produced almost identical results.
The effect of grid refinement was also investigated and it appeared that the results on a coarse grid (42*42) did not differ significantly from a finer grid (200*200). So for this type of problems we may conclude that a coarse grid can be used because grid refinement does not leed to more accurate results. This is cheaper with respect to CPU-time.

Splashing droplets:
Three cases were studied and compared to Rein[3] who produced photographs of the three experiments. This time the differences were bigger than in the case of splatting droplets.
The formation of a vortex ring: SAVOF96 had difficulties when it came to produce a clearly visible vortex ring. It is not obviously clear how the deformation takes place; little can be seen on the velocity/vorticity plots.
The formation of a central jet: The results were not as smooth as seen on the photographs but more 'wild'.
The formation of a crown: This was handled very well by SAVOF96 and quite resembling results were produced.

grid refinement: In contrary to splatting droplets, for splashing droplets grid refinement improved the results significantly. Much better and accurate results were achieved when using a 200*200 grid instead of a 42*42 grid. So for this type of problems a fine grid is recommended because otherwise important details will be missed.

Postprocessing: CPU-time and visualization:
Calculations on a coarse grid took only a few minutes CPU-time while the CPU-time on fine grids was approximately 2 hours.
The postprocessing and visualization in MATLAB on the other hand was a very tedious, time-consuming and expensive task. The plotting and converting part is worth adjusting in order to lower the costs for this process.

We may conclude that for the cases described in this paper SAVOF96 produced reasonable solutions for the Navier-Stokes equations where for splatting droplets the results were better than for splashing droplets .

Appendix A

Program description

This appendix contains a practical description of SAVOF96. The extended version of SAVOF96 consists of over 6000 lines of FORTRAN 77 code. The first section describes SAVOF96's calling sequence. Then some important common block variables are discussed and a brief explanation of the subroutines, input- and output files is given.

A.1 Calling sequence

The main calling sequence —excluding postprocessing routines — within SAVOF96 can be represented as follows:

```
initialisation      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
```

There is a routine PRT that is invoked every prtdt/delt time steps, to be specified in the input file. This occurs after VFCONV is executed. Then PRT calls all the desired subroutines that produce specific output. These and the other subroutines are discussed in appendix A.3.

A.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.

/CASE/ contains external body forces and their controls:

Omega, DelOme	: rotation rate and initial rotation relative to Omega.
X0, Y0	: in the case of two-dimensional rotation: point about which the rotation takes place.
TwUp, TwDown	: time to start and end the rotation.
U0, V0	: initial velocities in x- and y-direction respectively.
Ampl, Freq, Aangle	: amplitude, frequency and the angle under which an oscillation is performed.
GX, GY	: accelerations in x- and y-direction respectively.
TxOn, TxOff, TyOn, TyOff	: times to turn accelerations in x- and y-direction on and off.

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

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

/GRIDAR/ contains parameters involving gridsize:

X(I)	: x-coördinates of the grid-points.
XI(I)	: x-coördinates of the 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-coördinates of the grid-points.
YJ(J)	: y-coördinates of the 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
RDX(I)	: i.e. $\frac{1}{x_i}$, $\frac{2}{x_{i-1}+x_i}$ and $\frac{1}{\Delta x_i}$.
Circum(I)	: the circumference of a circle with radius x_i (1 in the axi-symmetric case , 0 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,JMaxUs	: number of grid points in x-and y-direction.
IM1Us, JM1Us,	: IMaxUs=IM1Us+1=IM2Us+2 and
IM2Us, JM2Us	: JMaxUs=JM1Us+1=JM2Us+2.

/ORGA/ is used for cell labeling:

NF(I,J),	: contains cell labels for the current and previous time
NFN(I,J)	: level respectively (see page 9).
PETA(I,J)	: coëfficient used to interpolate the pressure (see page 11).

/PHYS/ contains pressures and velocities at the current and the previous time level. (An N means at the previous time level)

F(I,J), FN(I,J)	: VOF (volume of fluid) / indicator function (see page 9).
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 the free surface.
VMAX	: maximum attained velocity.
PMIN, PMAX	: minimum and maximum pressure.

The latter three numbers 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 timestep.
TFin	: end time.
TStart	: start time.

A.3 Subroutines

In this section a short description of SAVOF96's subroutines is given.

BC : sets boundary conditions for velocity components.

BCBND : boundary conditions at outer boundary.

BDYFRC: computes the apparent body force.

CFLCHK: monitors CFL number and sets flag for time-step adjustment

COEFF : defines coefficient matrix for Poisson equation including boundary conditions at wall and free surface.

CONVRT: converts 2-dim datastructure into 1-dim datastructure for more efficient implementation of pressure solver.

DTADJ : halves/doubles time step (old time step will be repeated).

GRID : makes (non-uniform) grid.

INIT : starts new time step.

LABEL : labels empty, surface (preliminary) and full cells.

LDSTAT: reads restart file (produced by SVSTAT).

MILU : solves Poisson equation through 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: initializes fluid configuration.

SETPAR: reads input file.

SLAG : performs one SOR sweep. It uses a 1-dimensional implementation.

SOLVEP: organizes pressure calculation and updates velocity.

SURDEF: generates liquid configuration.

SVSTAT: writes restart file.

TILDE : integrates momentum equations.

VFCONV: moves fluid, i.e. adjusts VOF-function, and re-labels

AVS, GNUPLT, MATLAB, etc: generate plots and plotfiles.

A.4 Program adjustments

In order to produce the plots of the maximal splat radius R(t) and the pressure p as function of the time t on page .., a few lines had to be added to the main program SAVOF96.F. The adjustments were made in the subroutine PRT as shown below.

```
c** straal berekenen
         DO I=IM1Us,2,-1
            DO J=2,JM1Us
               IF (NF(I,J).NE.6) THEN
                                    FMAX=0.0
                                    DO J2=2,JM1Us
                                       IF (F(I,J2).GT.FMAX) FMAX=F(I,J2)
                                    ENDDO
                                    STRAAL=X(I-1)+FMAX*DelX(I)
                                    GOTO 30
               ENDIF
            ENDDO
         ENDDO
 30       CONTINUE
```

A.5 Input files

SAVOF96 uses at least one input file, the main input file SAVOF96.IN. It has the following structure:

```
********************************************************************************
SAVOF96-2.0 waterdruppel met radius 0.0345 cm en botsingssnelheid 1.46 m/sec
***** tank geometry **************************************************
icyl     Xmin      Xmax     Ymin     Ymax    SpGeom  SpMot
  1       0.0       0.1      0.0      0.1       0      0

***** liquid configuration *******************************************
LiqCnf   xp    yp     r     xq     yq
   2     0.0   3.45e-2   3.45e-2  0.0    0.0

***** grid definition ************************************************
iMaxUs   jMaxUs    cx    cy    Xpos  Ypos
 300     300      0.0   0.7    0.0    0.0

***** liquid properties **********************************************
Sigma    CAngle     Nu
73      90.0    1e-2

***** body forces and external motion: 2D ****************************
Gx   TxOn      TxOff      u0       Gy     TyOn     TyOff    v0
0.0      0.0       1000   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      100.0     0.0     0.0

***** body forces and external motion: axisymmetric ******************
     Gy    TyOn    TyOff   v0    Ampl    Freq
 -980     0.0   100.0     -146  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   UIn  VIn   Freqin  IPIN  PIn
   2        2      2       2   0.0  0.0     0.0   0     0

***** upwind parameter and Poisson iteration parameters **************
Alpha      Epsi     ItMax   OmStrt  IMilu
  1.0    3.0e-5       50      1.9    1

***** time step and restart control **********************************
TFin    DelT    CFLMax    PrtDt     svst   svdt
```

```
 .0008  0.000047  0.6        0.000047    1    50.0

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

***** stream lines ***************************************************
nrx    nry   nrdt     xps      yps     xqs      yqs       t   dt/delt
  0      0      0  135.0     30.0   150.0     50.0     0.0     4e4

***** fluxes *********************************************************
number of fluxes to be printed (flux01.out...flux**.out are created)
    0
   p1     p2     p3     hor   (line from (p1,p3) to (p2,p3) if hor=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))

 ---------------------------------------------------------------------
***** E X P L A N A T I O N ************* E X P L A N A T I O N *****
 ---------------------------------------------------------------------
**TANK GEOMETRY**
icyl    :  0=2-dim; 1=axisymmetrical

SpGeom  : reads programmed geometry savof96.geo*

**LIQUID CONFIGURATION**
LiqCnf  :    1=lower part of cylinder with semi-circle - avg. height yp;
             2=toroide (drop) - mpt cirkel (xp,yp), radius r;
             3=rectangle - xp < x < xq and yp < y < yq;
             4=drop along axis (y=yp, r) falling in waterpool deep yq;
             5=fluid filament - width r, height yp+semi-sphere

**GRID**
iMaxUs (<=130),jMaxUs (<=130): number of cells inclusive mirror cells
cx, cy     : stretchparameters - 0=no stretch;>0=smaller cells near position
              indicated by x(y)pos; <0=smaller cells away from x(y)pos

**LIQUID PROPERTIES**
Sigma      : kimematic surface tension (sigma/rho)
CAngle     : contact angle
Nu         : kinematic viscosity (mu/rho)

**EXTERNAL MOTION 2D**
```

```
Gx           : acceleration in horizontal direction
TxOn,TxOff : timeinterval where acceleration Gx is active (TxOn < t < TxOff)
u0           : horizontal initial velocity

Gy           : acceleration in vertical direction
TyOn,TyOff : timeinterval where acceleration Gy is active (TyOn < t < TyOff)
v0           : vertical initial velocity

Ampl,Freq,Angle : oscillation under angle with x-axis

Rpm          : rotation (per minuti) in (x,y)-plane
DelOme       : relative initial rotation
TwUp,TwDown: timeinterval where rotation is increased and decreased
x0,y0        : center of rotation

**EXTERNAL MOTION AXISYMMETRIC**
Gy           : acceleration in vertical direction
TyOn,TyOff : timeinterval where acceleration Gy is active (TyOn < t < TyOff)
v0           : vertical initial velocity

Ampl, Freq : oscillation in vertical direction

Rpm          : rotation (per minute)
DelOme       : relative initial rotation
TwUp,TwDown: timeinterval where rotation is increased and decreased

    [0 (spin-up)  Tup (maximum) Tdown (spin-down) Tdown+Tup (rest) TFin]

**BOUNDARY CONDITIONS**
bc           : 1=slip; 2=no-slip; 7=outflow; 8=inflow
UIn, VIn, FreqIn: inflow specification
IPIn, PIn  : boundary and value of pressure condition

**NUMERICAL PARAMETERS**
Alpha        : upwind parameter - 0=centraal; 1=upwind
Epsi         : Poisson convergence criterion
ItMax        : maximum number of iterations
OmStrt       : initial relaxation factor

**TIME CONTROL**
TFin         : endtime
DelT         : timestep
CFLMax       : maximum allowed CFL number
PrtDt        : time between 2 small printouts
20*PrtDt     : time between 2 large printouts
svst         : save state for restart control
```

```
             0= no restart backup;
             1= saves restart file every svdt time-units;
             2= starts with restart file, and proceeds like 1
svdt       : time between saving of restart file

**PRINT/PLOT CONTROL**
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
********************************************************************************
```

A.6 Output files

SAVOF96 can produce many output files containing all sorts of information about the calculation. We shall discuss the most important ones briefly.

- **SAVOF96.OUT** This is SAVOF96's main output file. It contains the coördinates of the grid points, rough plots of the liquid inside the geometry at equidistant pints in time (specified in the main input file) and comments from SAVOF96 about the convergence evolution of the time step, etc. It also contains lines with the point 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.

- **UVPF****.DAT** These files contain information about the velocity, pressure and the liquid configuration. It is used for post-processing (with MATLAB).

- **SAVOF96.RST** This is SAVOF96's restart file. The file can reach considerably proportions if the number of grid points is large.

Bibliography

[1] J.Fukai et al. *Modeling of the deformation of a liquid droplet impinging upon a flat surface.* Physics of Fluids A 5(11), November 1993, pp.2588-2599.

[2] J.Fukai et al. *Wetting effects on the spreading of a liquid droplet colliding with a flat surface:Experiment and modeling.* Physics of Fluids 7(2), February 1995, pp.236-247.

[3] M.Rein. *The transitional regime between coalescing and splashing drops.* Journal of Fluid Mechanics 306, 1996, pp.145-165.

[4] F.Rodriguez and R.Mesler. *Some drops don't splash.* Journal of Colloid and Interface Science 106 (no.2), August 1985.

[5] A.E.P. Veldman. *Numerieke stromingsleer.* RijksUniversiteit Groningen. Lecture notes. March 1994.

[6] B.de Groot. SAVOF*96-Simulation of free-surface liquid dynamics in moving complex geometries.* RijksUniversiteit Groningen. Master's thesis. August 1996.

List of Figures