THE UNIVERSITY OF READING

A Marker and Cell Solution of the Incompressible Navier-Stokes Equations for Free Surface Flow

by

Ben Weston

Numerical Analysis Report 6/2000

The University of Reading
P O Box 220
Reading RG6 6AX
Berkshire, UK

DEPARTMENT OF MATHEMATICS

Dissertation for the MSc 'Numerical Solution of Differential Equations'
University of Reading 1999-2000

A Marker and Cell Solution of the Incompressible Navier-Stokes Equations for Free Surface Flow

Ben Weston

September 2000

Abstract

This study presents a clear and simple algorithm for the solution of a free surface problem by solving the Navier-Stokes equations in an Eulerian framework with finite difference and using a pressure correction philosophy. The fluid interface is captured using a Marker and Cell (MAC) method. The problem solved is the collapse of a fluid column using different marker particle spacing, and shows the reflection off the boundary. It is found that the problem can only be solved successfully with a marker particle spacing of 0.01. The model shows numerical instability when there is large deformation in the flow even when using adaptive timestepping. This is probably due to the Poisson equation for the pressure being solved with insufficient accuracy. The results for the reflection of the collapsed column off the wall show that the results could be improved with further refinement in the mesh.

Contents

List of Symbols

f_x	-	x component of body force
g_x	-	x component of gravity
n	-	Unit normal vector of the interface
m	-	Unit tangential vector of the interface
$p\dot{q}$	-	Volumetric heating of element
t	-	Time
u	-	x - component of velocity
$\tilde{u}$	-	Intermediate x – component of velocity
u_p	-	Two dimensional linearly interpolated x – component of velocity
v	-	y - component of velocity
w	-	z - component of velocity
x_p^n	-	Massless marker particle's x – coordinate
y_p^n	-	Massless marker particle's y – coordinate
C	-	Scalar Volume Fraction function
D	-	Divergence of the fluid
Fr	-	Froude number
Re	-	Reynold's number
T	-	Temperature
V	-	Velocity vector
V	-	Velocity of fluid in direction concerned
Z	-	Height of fluid from a datum
ρ	-	Density of fluid
Φ	-	Scalar Level Set function
τ	-	Normal and shear stress in the plane concerned
λ	-	Secondary viscosity co-efficient
μ	-	Viscosity co-efficient
Ψ	-	Potential function
ϕ	-	Pressure within the fluid
$\tilde{\phi}$	-	Intermediate pressure within the fluid
ϕ_s	-	Pressure at the fluid interface
ϕ_{ns}	-	Fluid surface pressure from the normal stress condition
ϕ_a	-	Additional outside pressure
ϕ_f	-	Current value of pressure in neighbouring cell
η	-	Co-efficient for pressure interpolation

1.0 Introduction

In Computational Fluid Dynamics the Navier-Stokes equations are often used as the governing equations for fluid flow calculations. These equations are used to model viscous incompressible flow in a wide range of problems. The incompressible Navier-Stokes equations can be written as

The Continuity Equation

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0 \qquad (1.1)$$

The Momentum Equations

x - component :
$$\frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho u V)=-\frac{\partial \phi}{\partial x}+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{yx}}{\partial y}+\frac{\partial \tau_{zx}}{\partial z}+\rho f_x \qquad (1.2a)$$

y - component :
$$\frac{\partial(\rho v)}{\partial t}+\nabla\cdot(\rho v V)=-\frac{\partial \phi}{\partial v}+\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{zy}}{\partial z}+\rho f_y \qquad (1.2b)$$

z - component :
$$\frac{\partial(\rho w)}{\partial t}+\nabla\cdot(\rho w V)=-\frac{\partial \phi}{\partial z}+\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{yz}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z}+\rho f_z \qquad (1.2c)$$

Where ρ is density of the fluid, V is the velocity of the fluid, u,v,w are the velocity components, ϕ is pressure, ρf_x is the body force acting on the fluid and τ_{xx} are the shear and normal stresses on the fluid.

These equations or variants of these equations have been solved successfully in both Eulerian and Lagrangian frameworks using finite differences, finite elements and spectral or pseudo-spectral methodologies. Eq. 1.1 and Eq. 1.2a-c are the time dependent formulation of the Navier-Stokes equations. Removal of the time derivative will result in the calculation of the steady state solution.

This study concentrates on using a finite difference representation of the equations in an Eulerian framework to solve a time dependent problem. The problem concerned in this study has the added complexity of a free surface.

Free surface representation is a large topic in itself, therefore a review of current methods is presented in Section 2. The problem to be solved is a dam break problem on a dry bed, the initial configuration of which is shown in Fig 1.1.

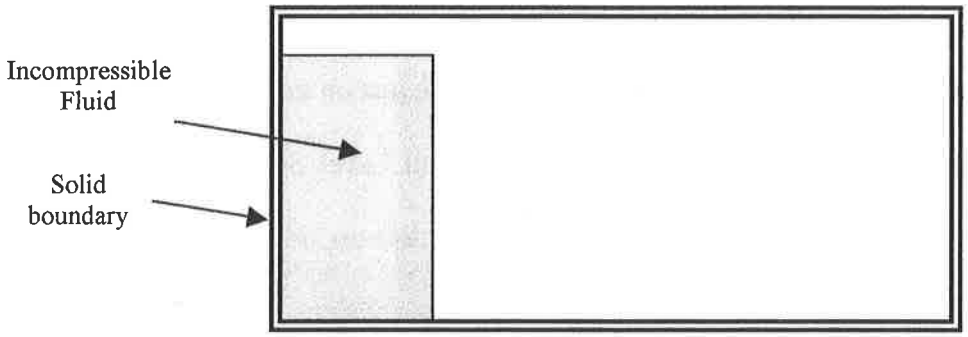


Fig 1.1 The initial configuration of the dam break problem at time t = 0.

This is a good problem to solve as it has a simple initial configuration and has been studied extensively with available experimental results. It is also a good problem to illustrate the modelling of a free surface since all orientations of the upper surfaces are represented. A Marker and Cell technique is used to track the free surfaces and this is outlined in Section 2. There are many ways of numerically solving the Navier-Stokes equations such as Poincare iterations and Newton-Raphson iterations but in this study we have chosen the pressure correction philosophy to solve the equations. Section 3 gives the details of the numerical procedure for the solution of the incompressible Navier-Stokes equations and the surface tracking method. Section 4 gives the results of the computations for different refinements of marker particles. Section 5 suggests further improvements to the model and summarises the main findings of the study.

2.0 Interface Capture Methods for Multi-Fluid Flow

In the numerical computation of many free surface and multi-fluid problems there must be an accurate representation of the interface between the fluids. The method must be able to track detached and merging fluids and have a sharply defined interface on which to accurately apply boundary conditions.

Currently there are two main methods for the computation of free surface flows

- Surface methods (direct methods)
- Volume methods (indirect methods)

Surface methods concentrate not on the whole fluid but just on the interface itself; this class of technique includes Particles on Interface and Level Set methods. Volume methods are applied over the whole computational domain, identify all fluids and attempt to approximate the interface shape between adjacent fluids; this class of technique includes Marker and Cell (MAC) and Volume Fraction methods.

2.1 Surface Methods

This class of method has the interface marked by special marker points, of which there are three main techniques.

2.1.1 Particles on Interface

Presented by Daly (1969) this involves marking the fluid interface with massless marker particles, as shown in Fig 2.1, which are advected with the local velocities of the fluid. This method is very sensitive to the number of marker particles used and the spacing between them.

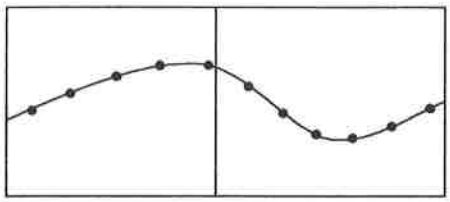


Fig 2.1 The Particle on Interface method

In order to apply the surface boundary conditions correctly the surface curvature must be known. This requires that the marker particles are sequentially numbered on the interface which means it is necessary to add and delete particles during the computation when they become unstructured. This is the major disadvantage of this method since when a fluid merges or detaches this sequential numbering is difficult to achieve.

2.1.2 Height Function

Presented by Nichols and Hirt (1973) this method is an extension of the Particles on Interface method in which the marker points are referenced from a set point or plane as shown by Fig 2.2a.

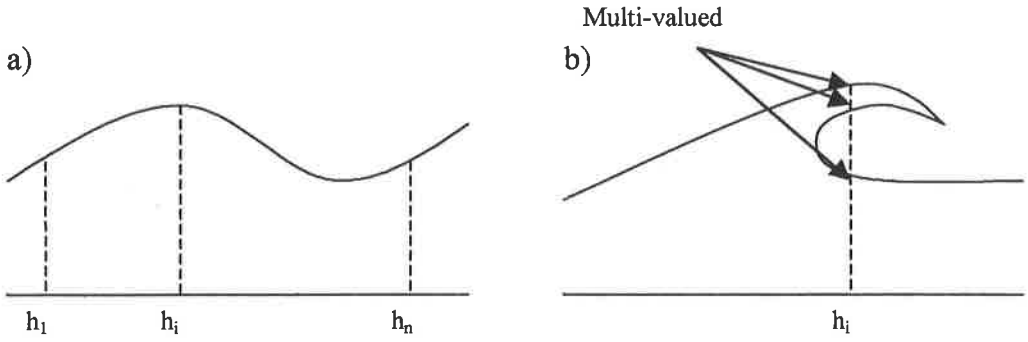


Fig 2.2 The Height function technique

Each height can only refer to one reference point and therefore it is impossible to model situations where the reference height must hold multi values such as a breaking wave as in Fig 2.2b

2.1.3 Level Set

This method involves the use of a level set function Φ (Osher and Sethian (1988)), over the whole computational domain. The Level Set function measures the distance away from the interface and is positive outside the fluid and negative inside and thus $\Phi = 0$ on the actual interface at shown by Fig 2.3.

The Level Set function is a scalar function, which is advected using a convection equation. The main drawback of this method is that at present there is no effective method of eliminating numerical diffusion from the advection process since Φ is not conservative.

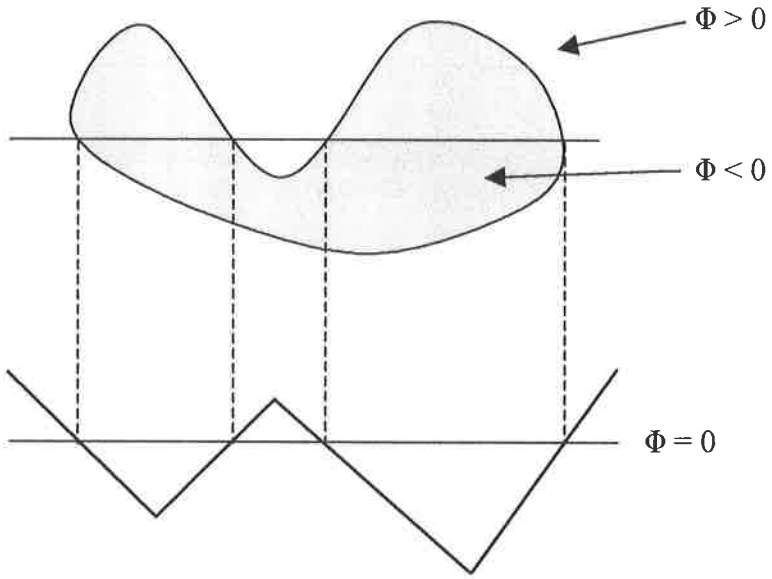


Fig 2.3 A random fluid region represented by the Level Set function, Φ

2.1.4 Surface Fitted Methods

This method fits the mesh to the surface as shown by Fig 2.4. The advantage of this method is that it cuts down on computer storage for marker cells, ensures a sharp interface and avoids partially filled cells, which means that surface boundary conditions can be applied accurately. The problem here lies in the fact that if the fluid

has large distortions then the fluid domain must be continually remeshed, which means added computational cost.

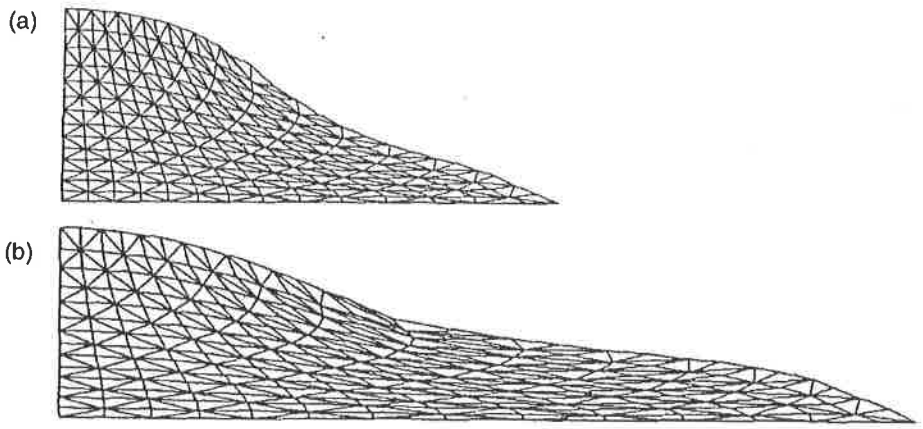


Fig 2.4 Collapse of a liquid column with a surface fitted method at time t = 3.0 and 4.0 from Ramaswamy & Kawahra, 1987, Int J. Numer. Methods Fluids, Vol 7 Fig 5

2.2 Volume methods

This class of methods is applied over the whole computational domain, where two or more fluids are identified. The following methods are then used to approximate the position of the interface between the separate fluids.

2.2.1 Marker and Cell Method (MAC)

This is an easy method to implement and involves the use of massless marker particles over the whole fluid domain (Harlow and Welsh (1965)). A cell flagging procedure is used over the domain, whereby a cell is either full, empty or a surface. A full cell initially is any cell with a marker particle present within it. An empty cell is any cell without a marker particle within it. A surface cell is any full cell adjacent to an empty cell. The MAC method can model highly deformed fluids and can also model detached and merging fluids. However it is heavy on computer storage, especially in three dimensions, as every marker particle's position must be stored.

2.2.2 Volume fractions

The volume fraction technique is simply a scalar function, C, which determines whether a cell is full, empty or partially filled. So if a cell is full C = 1, if it is empty C = 0, and if a cell is partially filled $0 < C < 1$, as shown by Fig 2.5.

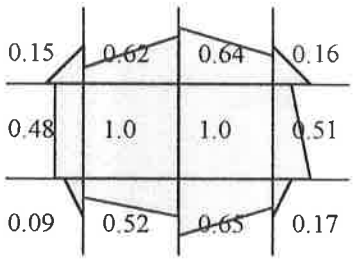


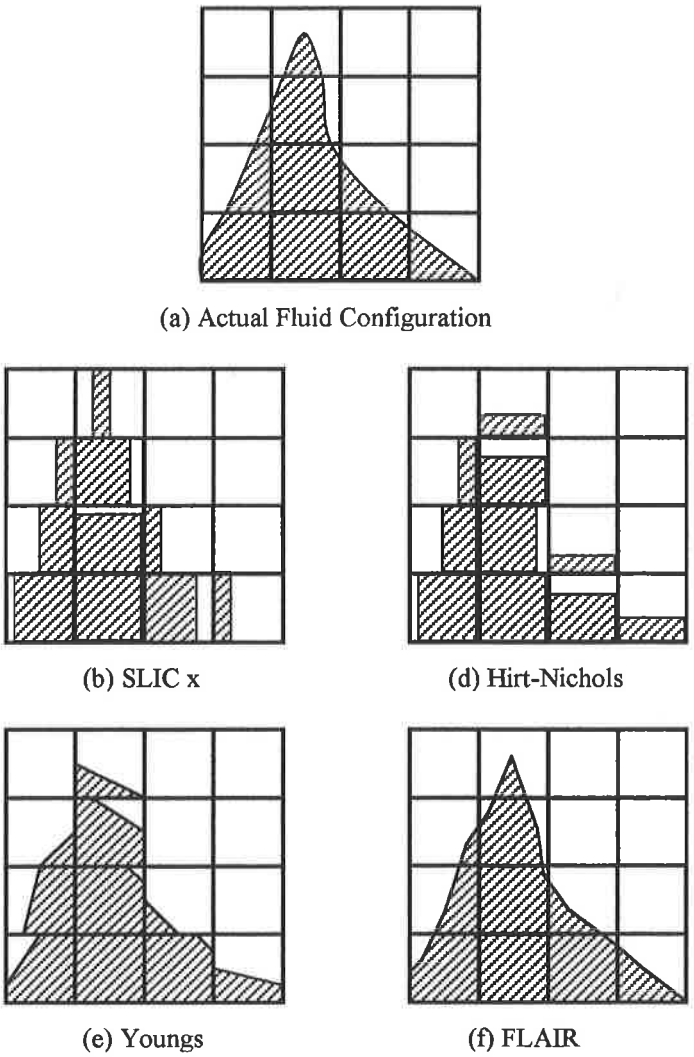
Fig 2.5 An example of the volume fraction function.

This volume fraction is then advected with the local velocities to update the fluid domain. Three main methods that have been considered are

- Line Techniques
- Donor-Acceptor Techniques
- Higher-Order Differencing Techniques

Line Techniques – There are many methods which fall into this category, e.g. SLIC (Noh and Woodward, 1976), Hirt and Nichols (Hirt and Nichols, 1981), Youngs' Method (Youngs, 1982), FLAIR (Ashgriz and Poo, 1991), HELMIT (Giddings, 1999). All these techniques represent the interface in a piecewise linear manner, and some of them are illustrated in Fig 2.6. The SLIC method reconstructs the interface parallel to either the x or y axis and uses an operator-split algorithm. The operator-split algorithm only uses information from neighbouring cells in the direction of the flux, i.e. an interface can have a different representation for the x and y directions. Once the reconstruction has been performed the fluxes are

calculated geometrically. Hirt-Nichols also uses a reconstruction that is parallel to either the x or y axis using a block of nine cells to calculate the interface. However in the Hirt-Nichols technique the interface is either vertical or horizontal depending on the magnitude of surface normal components. Youngs' method uses information about the normal of the surface to place a line according to the average normal of that cell.



(a) Actual Fluid Configuration

(b) SLIC x

(d) Hirt-Nichols

(e) Youngs

(f) FLAIR

Fig 2.6 Examples of Volume fraction interface reconstruction

FLAIR extends this idea by placing the interfaces on the cell faces. HELMIT extends these oblique line techniques further in that the intersection of a cell face by the

interface is a point halfway between its own intersection and that of its neighbouring cell, the additional volume being represented by an appropriately placed isosceles triangle (full details of this method can be found in Appendix B). All these techniques assume that the grid is rectangular to accommodate the operator-split method. If unstructured meshes are used then both the fluid advection and the interface reconstruction become very complex.

Donor – Acceptor Schemes – These schemes use volume fractions in the downwind cell to predict the volume of fluid transported through the relevant cell face. This method experiences instability problems resulting in smearing of the interface or the volume fraction becoming non-physical. Hirt and Nichols (1981) presented a Volume of Fluid (VOF) method based on this donor–acceptor scheme which tries to overcome some of these problems, with some success. However it can only manage to reduce them, but not to completely resolve the problem. This VOF method also uses the operator split method which assumes a rectangular grid.

Higher Order Differencing Schemes - This method is an attempt to minimise or resolve the difficulties encountered using the scalar convection equation. An example of this is STOIC (second and third order interpolation for convection) by Darwish (1993). This scheme is bounded and non-diffusive but fails to keep each step within one cell, resulting in other stability problems with the Navier-Stokes difference equations. Ubbink (1999) proposed a scheme CICSAM, an implicit bounded scheme which relies on a predictor-corrector type application. This prevents non-physical volume fractions occurring, and also can be applied to unstructured, non-quadrilateral meshes.

3.0 Approximate Solution of the Navier-Stokes Equations

In this section we give the governing equations for incompressible fluid flow and describe a specific numerical method that can be used to solve a range of incompressible fluid flows.

3.1 The Complete Navier Stokes Equations

The governing equations for three dimensional unsteady, viscous, compressible flow can be written in conservation form as

Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \tag{3.1}$$

Momentum Equations

$$x-\text{component}: \quad \frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u V) = -\frac{\partial \phi}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \tag{3.2}$$

$$y-\text{component}: \quad \frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v V) = -\frac{\partial \phi}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \tag{3.3}$$

$$z-\text{component}: \quad \frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w V) = -\frac{\partial \phi}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z \tag{3.4}$$

Energy Equation

(3.5)

$$\frac{\partial}{\partial t}\left[\rho\left(e + 1/2V^2\right)\right] + \nabla \cdot \left[\rho\left(e + 1/2V^2\right)\mathbf{V}\right] = \rho \dot{q} + \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)$$
$$- \frac{\partial(u\phi)}{\partial x} - \frac{\partial(v\phi)}{\partial y} - \frac{\partial(w\phi)}{\partial z} + \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} + \frac{\partial(v\tau_{zy})}{\partial z}$$
$$+ \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z} + \rho f \cdot \mathbf{V}$$

where ρ is density of the fluid, $\mathbf{V}$ is a vector of the fluid velocity, V is a scalar of the fluid velocity, u,v,w are the velocity components, ϕ is the pressure, ρf_x is the body force acting on the fluid, T is temperature, $p\dot{q}$ is the volumetric heating of the element, $1/2V^2$ is the kinetic energy per unit mass.

These equations are a coupled system of partial differential equations and are very difficult to solve analytically, with no closed form solution currently found with the exception of very special cases. The conservation form of these equations is often referred to as the divergence form. The normal and shear stress terms are functions of the velocity gradients and are defined as

$$\tau_{xx} = \lambda(\nabla \cdot \mathbf{V}) + 2\mu \frac{\partial u}{\partial x} \tag{3.6a}$$

$$\tau_{yy} = \lambda(\nabla \cdot \mathbf{V}) + 2\mu \frac{\partial v}{\partial y} \tag{3.6b}$$

$$\tau_{zz} = \lambda(\nabla \cdot \mathbf{V}) + 2\mu \frac{\partial w}{\partial z} \tag{3.6c}$$

$$\tau_{xy} = \tau_{yx} = \mu\left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right] \tag{3.6d}$$

$$\tau_{xz} = \tau_{zx} = \mu\left[\frac{\partial u}{\partial z} + \frac{\partial w}{\partial y}\right] \tag{3.6e}$$

$$\tau_{yz} = \tau_{zy} = \mu\left[\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right] \tag{3.6f}$$

where μ is the viscosity coefficient and λ is the secondary viscosity coefficient with λ = - 2/3 μ. When referring to a solution of the Navier-Stokes equations we take this to mean solving the full set of Navier-Stokes equations although Navier-Stokes equations just refer to the momentum equations.

3.2 The Incompressible Navier Stokes Equations

In the previous section we have summarised the full Navier Stokes equations. The incompressible form of the equations can be derived simply by setting density equal to a constant. Therefore with ρ = const Eq. 3.1 becomes

$$\nabla \cdot \mathbf{V} = 0 \tag{3.7}$$

and assuming μ = const, then Eq. 3.2-3.4 combined with Eq. 3.6a-f become

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u V) = -\frac{\partial \phi}{\partial x} + 2\mu \frac{\partial^2 u}{\partial x^2} + \mu \frac{\partial}{\partial y}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) + \mu \frac{\partial}{\partial z}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) + \rho f_x \tag{3.8}$$

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v V) = -\frac{\partial \phi}{\partial y} + \mu \frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) + 2\mu \frac{\partial^2 v}{\partial y^2} + \mu \frac{\partial}{\partial z}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) + \rho f_y \tag{3.9}$$

$$\frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w V) = -\frac{\partial \phi}{\partial x} + \mu \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) + \mu \frac{\partial}{\partial y}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) + 2\mu \frac{\partial^2 w}{\partial z^2} + \rho f_z \tag{3.10}$$

Note that all terms involving $\nabla \cdot \mathbf{V}$ can be set to zero by virtue of Eq. 2.7, thus from

$$\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{3.11}$$

we have by rearrangement

$$\frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \tag{3.12}$$

and then differentiating with respect to x we obtain

$$\frac{\partial^2 u}{\partial x^2} = -\frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 w}{\partial x \partial z} \tag{3.13}$$

Now adding $\frac{\partial^2 u}{\partial x^2}$ to both sides and multiplying by μ we obtain

$$2\mu \frac{\partial^2 u}{\partial x^2} = \mu \frac{\partial^2 u}{\partial x^2} - \mu \frac{\partial^2 v}{\partial x \partial y} - \mu \frac{\partial^2 w}{\partial x \partial z} \tag{3.14}$$

and if we substitute into Eq. 3.8 we obtain

$$\frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho uV)=-\frac{\partial\phi}{\partial x}+\mu\frac{\partial^2 u}{\partial x^2}-\mu\frac{\partial^2 v}{\partial x\partial y}-\mu\frac{\partial^2 w}{\partial x\partial y}+\mu\frac{\partial^2 v}{\partial x\partial y}+\mu\frac{\partial^2 u}{\partial y^2}+\mu\frac{\partial^2 u}{\partial z^2}+\mu\frac{\partial^2 w}{\partial x\partial y}+\rho f_x \tag{3.15}$$

Simplifying Eq. 3.15 gives

$$\frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho uV)=-\frac{\partial\phi}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)+\rho f_x \tag{3.16}$$

which can be written as

$$\frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho uV)=-\frac{\partial\phi}{\partial x}+\mu\nabla^2 u+\rho f_x \tag{3.17}$$

Applying a similar argument to the y and z momentum (Eq. 3.9-3.10) results in the incompressible Navier Stokes equations, written

Continuity : $$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{3.18}$$

x momentum : $$\frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho uV)=-\frac{\partial\phi}{\partial x}+\mu\nabla^2 u+\rho f_x \tag{3.19}$$

y momentum : $$\frac{\partial(\rho v)}{\partial t}+\nabla\cdot(\rho vV)=-\frac{\partial\phi}{\partial y}+\mu\nabla^2 v+\rho f_y \tag{3.20}$$

z momentum : $$\frac{\partial(\rho w)}{\partial t}+\nabla\cdot(\rho wV)=-\frac{\partial\phi}{\partial z}+\mu\nabla^2 w+\rho f_z \tag{3.21}$$

These 4 equations completely describe an incompressible flow and contain 4 dependent variables u,v,w and ϕ. Using the assumption that ρ and μ are constant the Energy equation has been completely decoupled. This means that the continuity and momentum equations are all that is required to solve for the velocity and pressure fields. If a given problem involves heat transfer, then the temperature field can be calculated from the energy equation after the velocity and pressure fields have been calculated.

3.3 Numerical Solution of the Incompressible Navier-Stokes Equations

Considering Eq. 3.15 and substituting

$$\nabla \cdot (\rho u V) = \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} \tag{3.22}$$

together with the assumption of two dimensional flow in the x-y plane,

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = -\frac{\partial \phi}{\partial x} + \mu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + \rho f_x \tag{3.23}$$

Defining the Reynolds number and the Froude number as

$$\mathrm{Re} = \frac{\rho}{\mu}, \qquad Fr = \frac{1}{\sqrt{gZ}} \tag{3.24}$$

we obtain the non dimensional form of (3.23) as

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial \phi}{\partial x} + \frac{1}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + \frac{1}{Fr^2} g_x \tag{3.25}$$

Applying the continuity equation to Eq. 3.25, differentiating Eq. 3.22 with respect to x and y, we obtain, respectively,

$$\frac{\partial^2 u}{\partial x^2} = -\frac{\partial^2 v}{\partial x \partial y}, \qquad \frac{\partial^2 v}{\partial y^2} = -\frac{\partial^2 u}{\partial x \partial y} \tag{3.26}$$

Direct substitution of these into Eq. 3.25 will yield Eq. 3.27-3.29, our basic governing equations for two dimensional time dependent incompressible flow in non dimensional form, i.e,

Continuity : $$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.27}$$

x momentum : $$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial \phi}{\partial x} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + \left(1/Fr^2\right) g_x \tag{3.28}$$

y momentum : $$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial \phi}{\partial y} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + \left(1/Fr^2\right) g_y \tag{3.29}$$

3.3.1 Staggered Gridding

Consider the following velocity distribution,

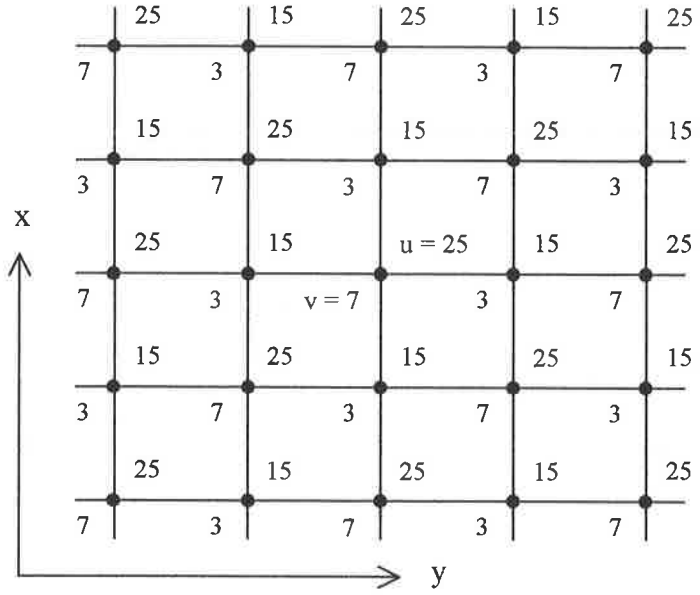


Fig 3.1 A checkerboard pattern for the velocity distribution

If we use central differences to approximate Eq. 3.27 (the continuity equation) we have

$$\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} = 0 \qquad (3.30)$$

The above checkerboard velocity distribution will now satisfy the continuity equation, but the above velocity distribution in Fig 3.1 is not physically possible. This problem is unique to incompressible flow, whereas if the flow is compressible then the density variation term present would correct this problem.

Similarly if we now consider the pressure distribution as shown in Fig 3.2 and represent the pressure derivative terms in the momentum equations using central differences, then we obtain

$$\frac{\partial \phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \qquad , \qquad \frac{\partial \phi}{\partial y} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y} \qquad (3.31)$$

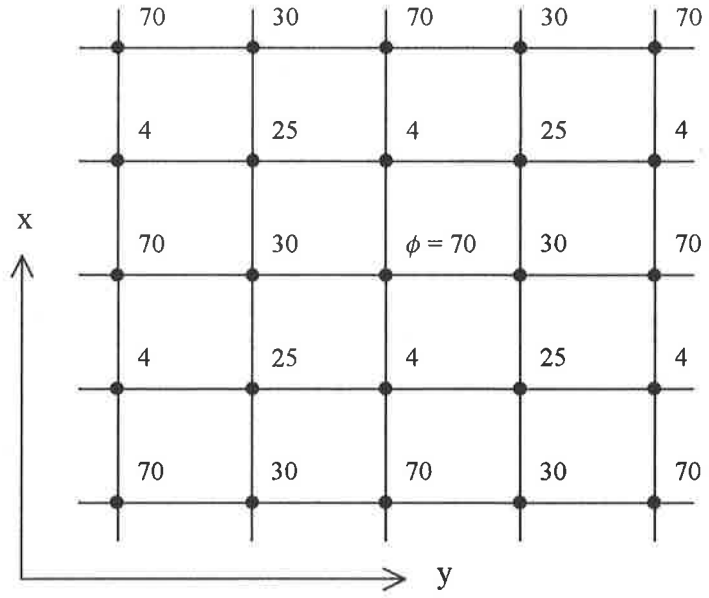


Fig 3.2 A checkerboard pattern for the pressure distribution

According to the pressure distribution shown in Fig 3.2 the pressure derivative term would be zero and the pressure distribution would not affect the velocity field, which again does not make physical sense.

An acceptable cure for this problem is to use a staggered grid, i.e. all the pressures are calculated at the solid grid points (i,j), (i+1,j), (i,j-1) etc, and the velocities are calculated at a translation of ± ½ Δx and ± ½ Δy, i.e. at the grid points (i+½,j), (i–½,j), (i,j+½), (i,j-½). The velocities and the pressures are now all calculated at different grid points, as shown by Fig 3.3.

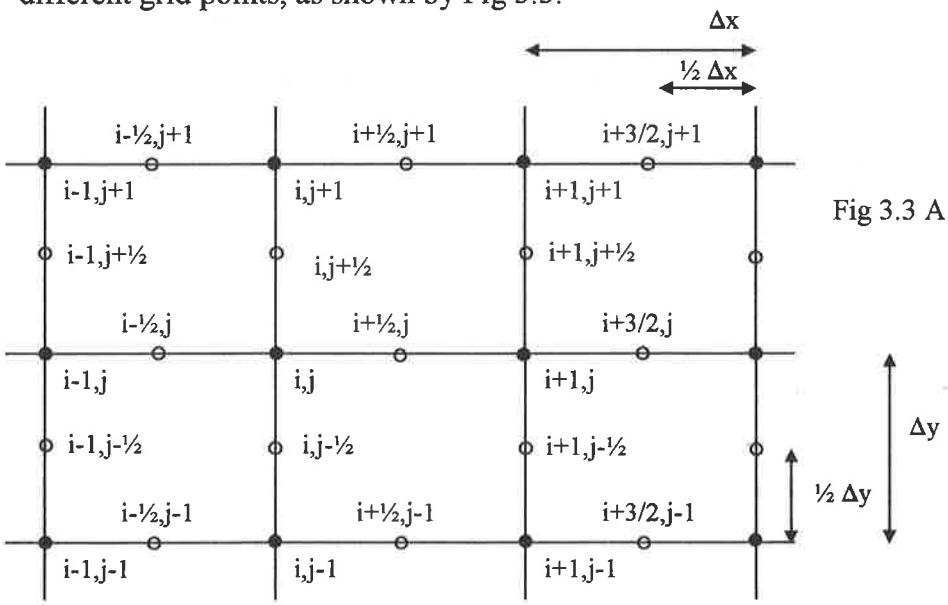


Fig 3.3 A staggered grid

The key advantage of this is that, for example, in the calculation of $u_{i+½,j}$ the pressure derivative can now be approximated as

$$\frac{\partial \phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \tag{3.32}$$

This eliminates the possibility of a checkerboard pressure distribution. Similarly the central difference approximation of the continuity equation would be

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \tag{3.33}$$

which again eliminates the possibility of a checkerboard pattern.

A point to note for later is that the translation $\sigma\Delta x$ where σ = ½ need not be uniformly ½ but the translation could be $0 < \sigma < 1$, as long as the information is kept.

3.3.2 Numerical Procedure : Pressure Correction

The method of solution for this non-linear partial differential equation is to linearise and solve on a staggered grid. The procedure can be described as follows. Initially let us assume we have a known velocity field, $u(x,t_o)$ at time t_o and that boundary conditions for the velocity and pressure field are given. The updated velocity field at $t = t + \delta t$ can be found as follows :

1. Let $\tilde{\phi}(x,t_o)$ be an arbitrary pressure field, which satisfies the correct boundary conditions at the free surface.
2. Calculate the intermediate velocity field $\tilde{u}(x,t)$, from

$$\frac{\partial \tilde{u}}{\partial t} = \left[-\frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} - \frac{\partial \tilde{\phi}}{\partial x} + \frac{1}{\mathrm{Re}} \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + \left(1/Fr^2 \right) g_x \right]_{t=t_o} \tag{3.34}$$

$$\frac{\partial \tilde{v}}{\partial t} = \left[-\frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} - \frac{\partial \tilde{\phi}}{\partial y} - \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + \left(1/Fr^2 \right) g_y \right]_{t=t_o} \tag{3.35}$$

with $\tilde{u}(x,t_o) = u(x,t_o)$ using the correct boundary conditions, We now define the true velocity field as

$$u(x,t) = \tilde{u}(x,t) - \nabla\Psi$$

with

$$\nabla^2\Psi = \nabla \cdot \tilde{u}(x,t)$$

so that u(x,t) satisfies,

$$\nabla \cdot u(x,t) = 0$$

3. Solve the Poisson equation,

$$\nabla^2\Psi = \nabla \cdot \tilde{u}(x,t) \tag{3.36}$$

4. Compute the velocity field,

$$u(x,t) = \tilde{u}(x,t) - \nabla\Psi \tag{3.37}$$

5. Compute the pressure field,

$$\phi = \tilde{\phi} + \frac{\Psi}{\delta t} \tag{3.38}$$

6. Advect the fluid and reconstruct the fluid interface.

3.3.3 The Finite Difference Equations

As with most Navier-Stokes calculations using finite differences, a staggered grid is employed, the variables being pressure $\tilde{\phi}_{ij}$, the added velocity potential Ψ_{ij} and the divergence D_{ij} which are positioned at the cell centre, while the velocities u_{ij} and v_{ij} are staggered by a translation of $\delta x/2$ and $\delta y/2$ respectively, as shown by Fig 3.4.

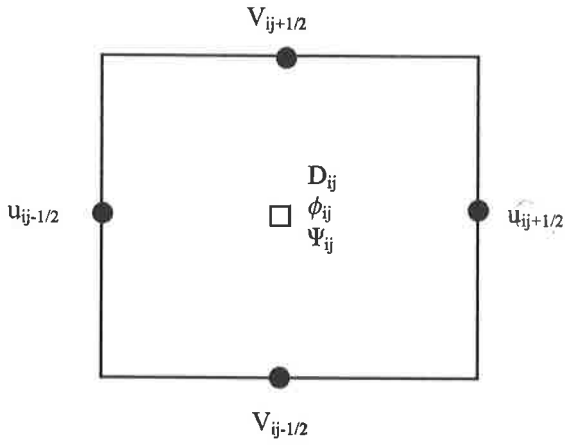


Fig 3.4 A typical cell in the computational domain

The momentum equations (Eq. 3.28-3.29) are discretised and applied to the u and v nodes separately. In finite difference form these equations become

$$\frac{\tilde{u}^{n+1}_{i+1/2,j}-\tilde{u}^{n+1}_{i+1/2,j}}{\delta t}=\frac{\tilde{\phi}_{i,j}-\tilde{\phi}_{i+1,j}}{\delta x} \tag{3.39}$$

$$+\frac{u^n_{i+1/2,j-1/2}v^n_{i+1/2,j-1/2}-u^n_{i+1/2,j+1/2}v^n_{i+1/2,j+1/2}}{\delta y}+\frac{u^n_{i+1/2,j}u^n_{i-1/2,j}-u^n_{i+3/2,j}u^n_{i-1/2,j}}{\delta x}$$

$$+\left(\frac{1}{\mathrm{Re}}\right)\left[\frac{u^n_{i+1/2,j+1}+u^n_{i+1/2,j-1}-2u^n_{i+1/2,j}}{\delta y^2}-\frac{v^n_{i+1,j+1/2}-v^n_{i+1,j-1/2}-v^n_{i,j+1/2}+v^n_{i,j-1/2}}{\delta y}\right]$$

$$+\left(1/Fr^2\right)g_x$$

$$\frac{\tilde{u}_{i,j+1/2}^{n+1} - \tilde{u}_{i,j+1/2}^{n+1}}{\delta t} = \frac{\tilde{\phi}_{i,j} - \tilde{\phi}_{i,j+1}}{\delta y} \tag{3.40}$$
$$+ \frac{u_{i-1/2,j+1/2}^{n} v_{i-1/2,j+1/2}^{n} - u_{i+1/2,j+1/2}^{n} v_{i+1/2,j+1/2}^{n}}{\delta x} + \frac{v_{i,j+1/2}^{n} v_{i,j-1/2}^{n} - v_{i,j+3/2}^{n} v_{i+1/2,j}^{n}}{\delta y}$$
$$- \left(\frac{1}{\mathrm{Re}}\right)\left[\frac{u_{i+1/2,j+1}^{n} - u_{i+1/2,j}^{n} - u_{i-1/2,j+1}^{n} + u_{i-1/2,j}^{n}}{\delta x \delta y} + \frac{2v_{i,j+1/2}^{n} - v_{i+1,j+1/2}^{n} - v_{i-1,j+1/2}^{n}}{\delta x^2}\right]$$
$$+ \left(1/Fr^2\right) g_y$$

The Poisson Equation (Eq. 3.36) is discretised using the 5-point Laplacian, which is written as

$$4\Psi_{ij} - \Psi_{i-1,j} - \Psi_{i,j+1} - \Psi_{i+1,j} - \Psi_{i,j-1} = -h^2 D_{ij} \tag{3.41}$$

where h = δx = δy and the divergence, D is

$$D_{ij} = \frac{\tilde{u}_{i+1/2,j} - \tilde{u}_{i-1/2,j}}{\delta x} + \frac{\tilde{v}_{i,j+1/2} - \tilde{v}_{i,j-1/2}}{\delta y} \tag{3.42}$$

3.3.4 Domain Boundary Conditions

The boundary conditions for the domain boundary are as follows :

$$\frac{\partial \Psi}{\partial n} = 0 \qquad \text{on domain boundary} \tag{3.43a}$$

$$\frac{\partial \phi}{\partial n} = 0 \qquad \text{on domain boundary} \tag{3.43b}$$

Free slip is imposed on all domain boundaries, which can be visualised as follows :

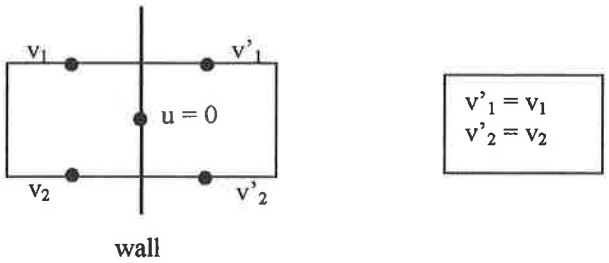


Fig 3.5 The representation of the free slip boundary condition

Hense the free slip has $u_{i+1/2,j} = 0$ at the wall surface and $v_{i,j+1/2} = v'_{i,j+1/2}$. Other orientations follow in a similar manner.

3.3.5 Free Surface Stress Conditions

Free surface boundary conditions are represented by

$$\phi - \frac{2}{\mathrm{Re}}\left[n_x n_x \frac{\partial u}{\partial x} + n_x n_y \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + n_y n_y \frac{\partial v}{\partial y} \right] = 0 \qquad (3.44)$$

$$\left[2 n_x m_x \frac{\partial u}{\partial x} + \left(n_x m_y + n_y m_x \right) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + 2 n_y m_y \frac{\partial v}{\partial y} \right] = 0 \qquad (3.45)$$

where $n = (n_x, n_y)$ the outward unit normal vector to the surface and $m = (m_x, m_y)$ is the unit tangential vector.

There are three main cases for implementation :

Case 1 : Surface cells with one side contiguous with empty cells.

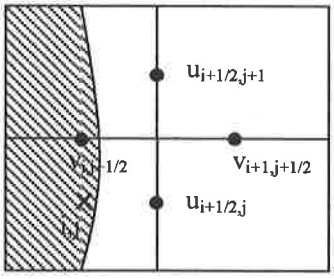


Fig 3.6 An example cell with one side contiguous with an empty cell

Here the assumption is that as the surface cuts through opposite sides of the cell, n_x or n_y will be small so that Eq. 3.44 and 3.45 can be represented by

$$\phi - \frac{2}{\mathrm{Re}}\left(\frac{\partial u_n}{\partial n} \right) = 0 \qquad \text{where n represents either x or y direction} \qquad (3.46)$$

$$\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \qquad (3.47)$$

Considering Fig 3.6 the finite difference representation of Eq.3.46 is

$$\phi_{i,j} = \frac{2}{\mathrm{Re}}\left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\delta x}\right) \tag{3.48}$$

Now the tangential stress can be represented by

$$v_{i+1,j+1/2} = v_{i,j+1/2} - \frac{\delta x}{\delta y}\left(u_{i+1/2,j+1} - u_{i+1/2,j}\right) \tag{3.49}$$

where $u_{i+1/2,j+1}$ and $u_{i+1/2,j}$ can be found by satisfying $\nabla \cdot \mathbf{V} = 0$.

Case 2 : Surface cells with two sides contiguous with empty cells.

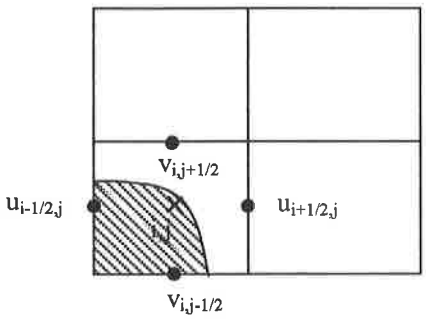


Fig 3.7 An example cell with two sides contiguous with an empty cell

For these cells we assume that the outward normal is at 45 degrees and therefore Eq. 3.44 and 3.45 become

$$\phi = \pm\frac{1}{\mathrm{Re}}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad \text{where the sign is the sign of } n_x n_y \tag{3.50}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.51}$$

For example a finite difference representation to the normal stress would be

$$\phi_{i,j} = \frac{1}{2\mathrm{Re}}\left[\frac{u_{i+1/2,j} + u_{i-1/2,j} + u_{i+1/2,j-1} + u_{i-1/2,j-1}}{\delta y} + \frac{v_{i,j+1/2} + v_{i,j-1/2} + v_{i-1,j+1/2} + v_{i-1,j-1/2}}{\delta x}\right] \tag{3.52}$$

For the tangential stress $\partial u/\partial x$ and $\partial v/\partial y$ must vanish separately. A finite difference representation of this would be

$$u_{i+1/2,j} = u_{i-1/2,j}\ , \qquad v_{i,j+1/2} = v_{i,j-1/2}$$

Case 3 : Surface cells with three sides contiguous with empty cells.

These cells should not really occur as it is an indication of a poorly refined grid. However if they do occur then one velocity should be adjusted so that $\nabla \cdot \mathbf{V} = 0$ is satisfied.

3.4 Adaptive Timestepping

The computational timestepping involves the use of adaptive timestepping. The chosen timestep is dependent on stability and accuracy requirements. Markham and Procter stated that for numerical stability the fluid should not be able to cross more than one cell boundary in a given time interval, that is,

$$|u|\delta t < \delta x \qquad , \qquad |v|\delta t < \delta y \tag{3.53}$$

Following Markham and Proctor the adaptive timestepping procedure can be defined as

$$\delta t_{visc} = 2\,\mathrm{Re}\cdot A_1 \cdot \frac{\delta x^2 \delta y^2}{\delta x^2 + \delta y^2} \tag{3.54}$$

$$\delta t_u = A_2 \cdot \frac{\delta x}{2U_{\max}} \tag{3.55}$$

$$\delta t_v = A_2 \cdot \frac{\delta y}{2V_{\max}} \tag{3.56}$$

where $0 \le A_i \le 1$. The time step used is now calculated following

$$\delta t = MIN\{\delta t_{visc}, \delta t_u, \delta t_v\} \cdot A \tag{3.57}$$

where $0 < A \le 1$. In Eq. 3.54-3.56 reference is made to U_{max} and V_{max}, If $\underset{i,j}{MAX}\left|U_{i,j}^{n+1}\right|$ were to be used this would drastically increase computational time. Instead a compromise is made and $\underset{i,j}{MAX}\left|\tilde{U}_{i,j}\right|$ is chosen. Now if δt_u or δt_v is less than δt then the timestep is

revised and $\tilde{U}_{i,j}$ is recalculated. The factor A is used to compensate for this use of $\underset{i,j}{MAX}\left|\tilde{U}_{i,j}\right|$ rather than $\underset{i,j}{MAX}\left|U_{i,j}^{n+1}\right|$.

3.5 Solution of the Poisson Equation

The discretised Poisson equation Eq. 3.41 leads to a linear system,

$$A\underline{x} = \underline{b} \tag{3.58}$$

The matrix A is formed by applying Eq. 3.41 to each full cell from right to left, whilst $\underline{b}$ is a vector of the divergence $\left(\nabla \cdot \tilde{D}\right)$ calculated at the same time. Once the A matrix is assembled then $\underline{x}$ represents the vector of the values of the potential function, Ψ. With the inclusion of the free surface in the problem then the A matrix can be shown to be symmetric and positive definite.

There are two main points to consider when solving the linear system Eq. 3.58, namely how the matrix can be stored and how it can be solved efficiently. Now, as the A matrix is sparse and symmetric and positive definite an efficient linear solver would be the method of conjugate gradients and an effective method of storing the non zeros would be Compressed Sparse Row. Compressed Sparse Row involves the use of three vectors to represent A and is best explained in terms of an example. Let us consider the matrix M of order, $n = 5$ and with the number of non zeros, $nz = 12$

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 3 & 4 & 0 & 5 \\ 6 & 0 & 7 & 0 & 0 \\ 0 & 8 & 0 & 9 & 10 \\ 0 & 0 & 11 & 0 & 12 \end{pmatrix}$$

and let us represent it using three vectors $\underline{A}(nz)$, $\underline{Col}(nz)$, $\underline{RPt}(n+1)$. $\underline{A}(nz)$ is real and contains the values of the non zeros in M. $\underline{Col}(nz)$ is integer and contains the

number of the column for each of the non zeros which are stored in $\underline{A}(nz)$. $\underline{RPt}(n+1)$ is integer and points to the position in the $\underline{Col}(nz)$ vector that each row starts: it assumes that in $\underline{Col}(nz)$ the rows are in order 1,2,....n and that the final entry is a flag ending the vector which takes the value $\underline{RPt}(1)+nz$. Now the three vectors should hold the following values to represent M,

$$A \equiv [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12]$$

$$Col \equiv [1 \quad 5 \quad 2 \quad 3 \quad 5 \quad 1 \quad 3 \quad 2 \quad 4 \quad 5 \quad 3 \quad 5]$$

$$RPt \equiv [1 \quad 3 \quad 6 \quad 8 \quad 11 \quad 13]$$

In order to solve the linear system using conjugate gradients a matrix.vector operation is required and the details for this algorithm can be found in Appendix B. As the matrix A from Eq. 3.33 is symmetric and positive definite then this allows the use of the conjugate gradient method effectively with the advantage that the search directions will be orthogonal in the A inner product, the details of the algorithm used can be seen in Appendix A. Note that implementation of pre-conditioners would prove to be difficult as the order of the system is continually changing, however this is an area for further research.

3.6 Implementation of the MAC Technique for Interface Tracking

As described in Section 2.2.1 this technique is an easy volume technique to implement and involves tracking the movement of massless marker particles over the fluid domain with the local velocities of the fluid.

To implement this technique a system of cell flagging is used, that is if a cell is a boundary it is marked B, a Full cell marked F, an empty cell marked E and a full cell

adjacent to at least one empty cell is remarked as surface S. This is illustrated in Fig 3.8, although this is a much coarser example than is used in the computations.

B	B	B	B	B	B	B	B	B	B	B	B	B
B	E	E	E	E	E	E	E	E	E	E	E	B
B	S	S	S	S	E	E	E	E	E	E	E	B
B	F	F	F	S	E	E	E	E	E	E	E	B
B	F	F	F	S	E	E	E	E	E	E	E	B
B	F	F	F	S	E	E	E	E	E	E	E	B
B	B	B	B	B	B	B	B	B	B	B	B	B

Fig 3.8 An example of a flagging system for a collapse of a liquid column problem

The marker particle co-ordinates are stored at each timestep and updated by solving

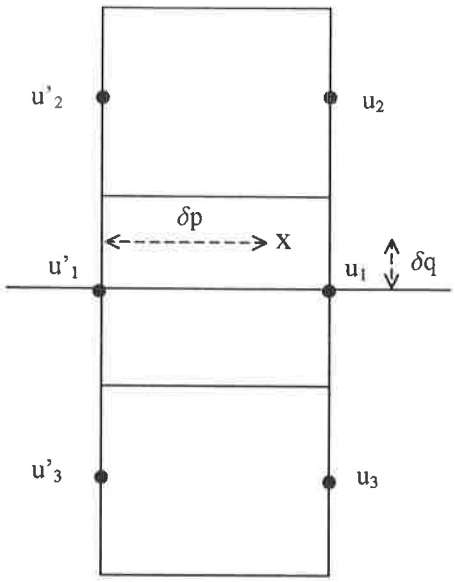
$$\frac{\partial x}{\partial t} = u \qquad \text{and} \qquad \frac{\partial y}{\partial t} = v \tag{3.59}$$

Once the velocity field has been calculated for the timestep then Eq. 3.59 are solved using Euler's method for the updated position of the particle. The new particle position can be found using Euler's method as

$$x_p^{n+1} = x_p^n + u_p \delta t \tag{3.60a}$$

$$y_p^{n+1} = y_p^n + v_p \delta t \tag{3.60b}$$

where $\left(x_p^n, y_p^n\right)$ is the current particle position and, $\left(x_p^{n+1}, y_p^{n+1}\right)$ is the particle's position at the next timestep. The velocities u_p and v_p are found using a weighting scheme which uses the four nearest velocities in a two dimensional linear interpolation. If we initially consider just u_p, then the linearly interpolated velocities for cells (i,j), ((i+1),j), and ((i-1),j) are found according to Fig 3.9.



$$\overline{u}_2 = \delta p u_2 + (1.0 - \delta p) u'_2$$

$$\overline{u}_1 = \delta p u_1 + (1.0 - \delta p) u'_1$$

$$\overline{u}_3 = \delta p u_3 + (1.0 - \delta p) u'_3$$

Fig 3.9 The equations of the linear interpolation weighting method

Now according to Fig 3.9 the interpolation in the y direction is

$$u_p = (1.0 - \delta q)\overline{u_1} + \delta q.\overline{u_2} \qquad \text{if } \delta q > 0$$

$$u_p = (1.0 + \delta q)\overline{u_1} - \delta q.\overline{u_3} \qquad \text{if } \delta q < 0$$

This weighting scheme generally works well and smooths out discontinuities in the velocity field resulting from a badly refined grid.

4.0 Results

4.1 Fluid Column Collapse with Marker Particle Spacing 0.02 units

Fig 4.1 shows the initial configuration of the fluid at time 0.0s, the massless marker particles being equally spaced at 0.02 units between each particle.

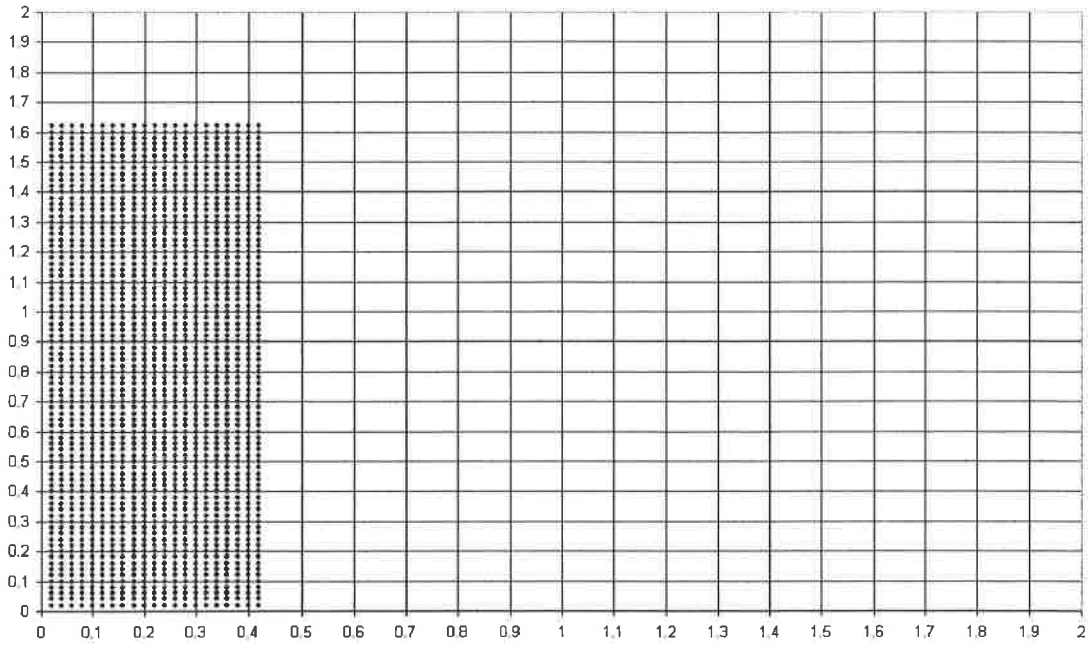


Fig 4.1 The initial configuration of the fluid at time 0.0s

Fig 4.2-4.7 show sample time frames at 0.2, 0.4, 0.6, 0.8, 1.0 and 1.1 respectively. These clearly show the column of falling fluid accelerating as time progresses. At time 1.1 the fluid reaches the domain boundary and begins to reflect off it, at which point the calculation was halted. The later frame times 1.0 and 1.1 begin to show numerical instability, probably due to the CFL condition being violated, and therefore the computation was stopped at this point.

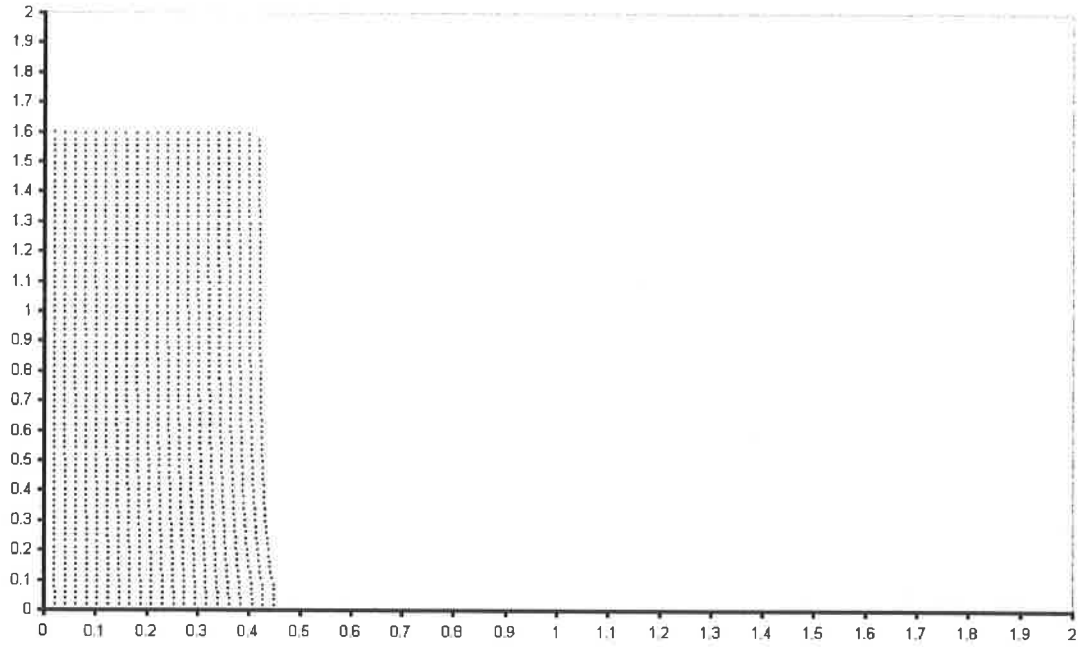


Fig 4.2 The evolution of the fluid at time 0.2

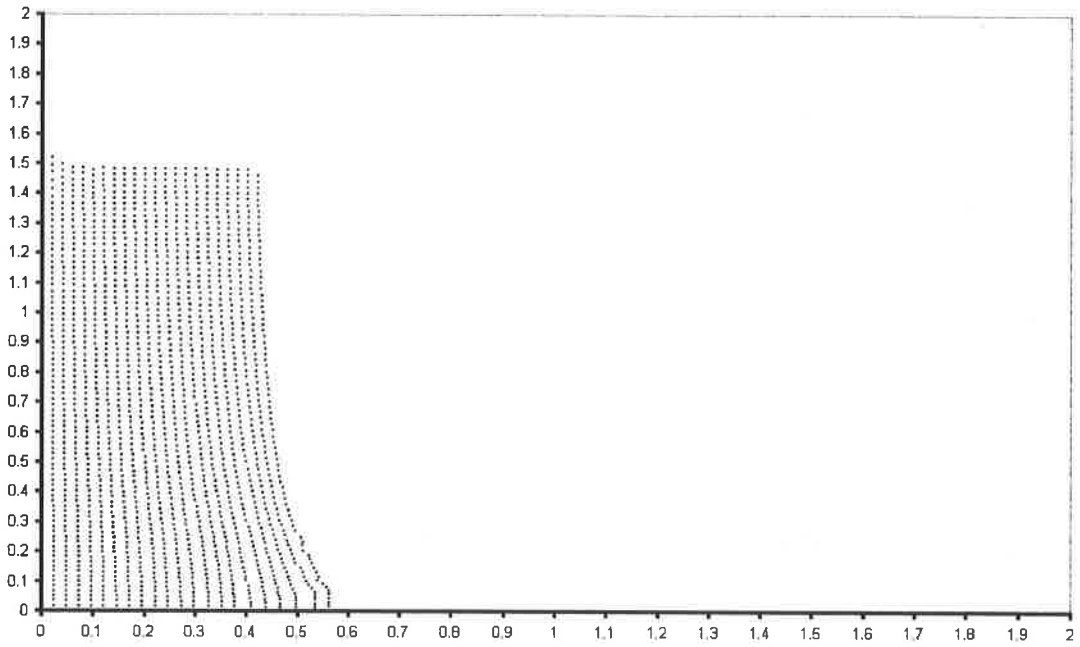


Fig 4.3 The evolution of the fluid at time 0.4

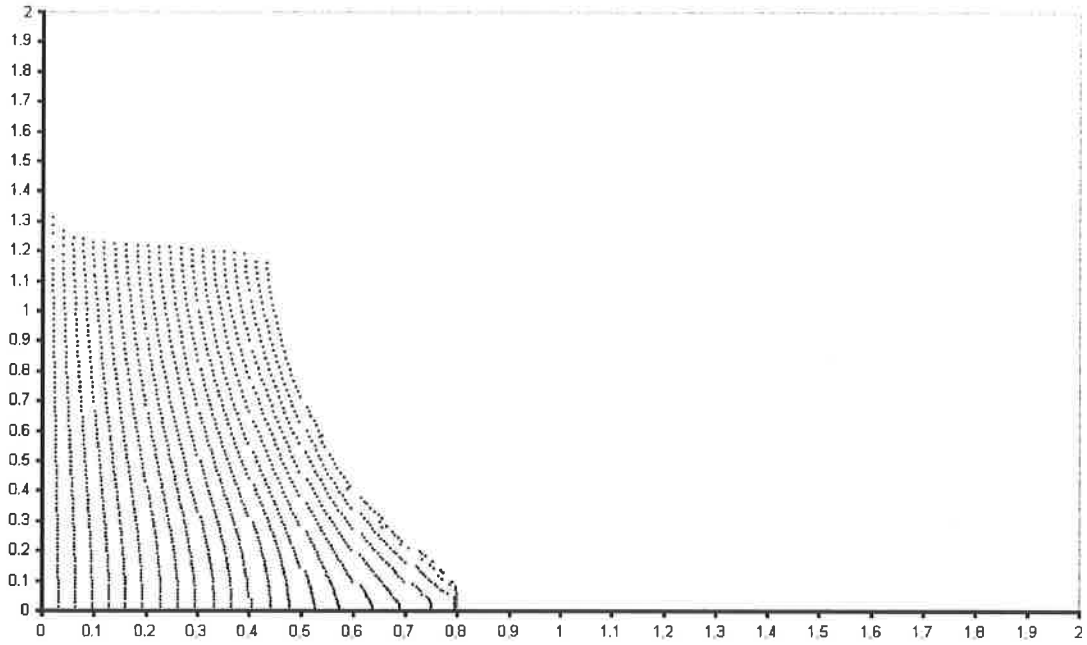


Fig 4.4 The evolution of the fluid at time 0.6

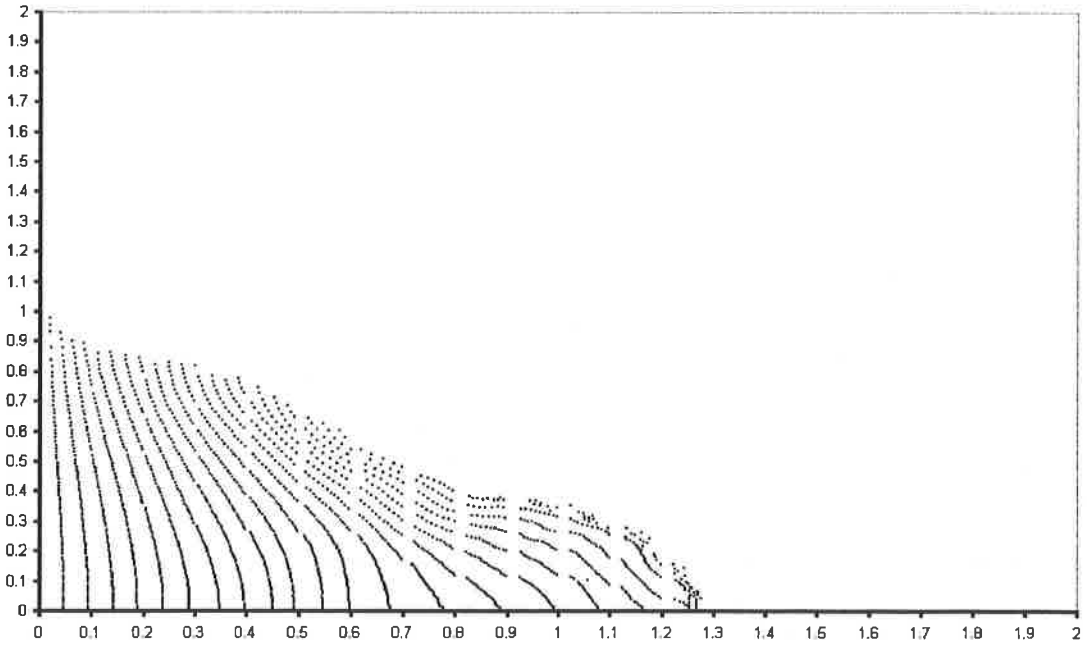


Fig 4.5 The evolution of the fluid at time 0.8

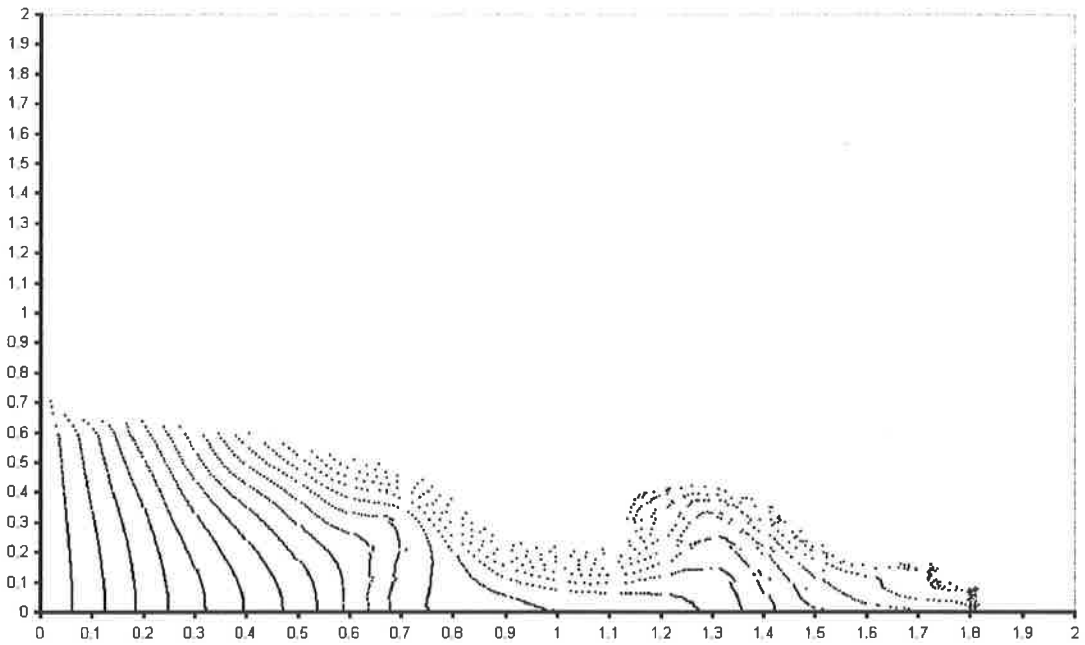


Fig 4.6 The evolution of the fluid at time 1.0

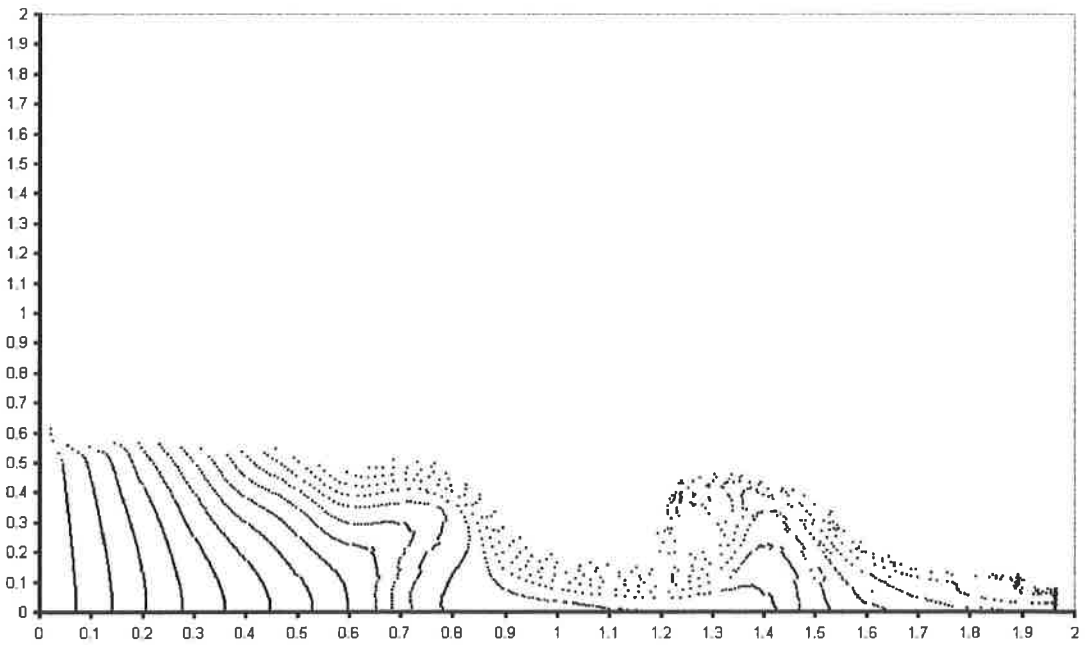


Fig 4.7 The evolution of the fluid at time 1.1

4.2 Fluid Column Collapse with Marker Particle Spacing 0.01 units

Fig 4.8 shows the initial configuration of the fluid at time 0.0s, the massless marker particles are now equally spaced at 0.01 units between each particle.

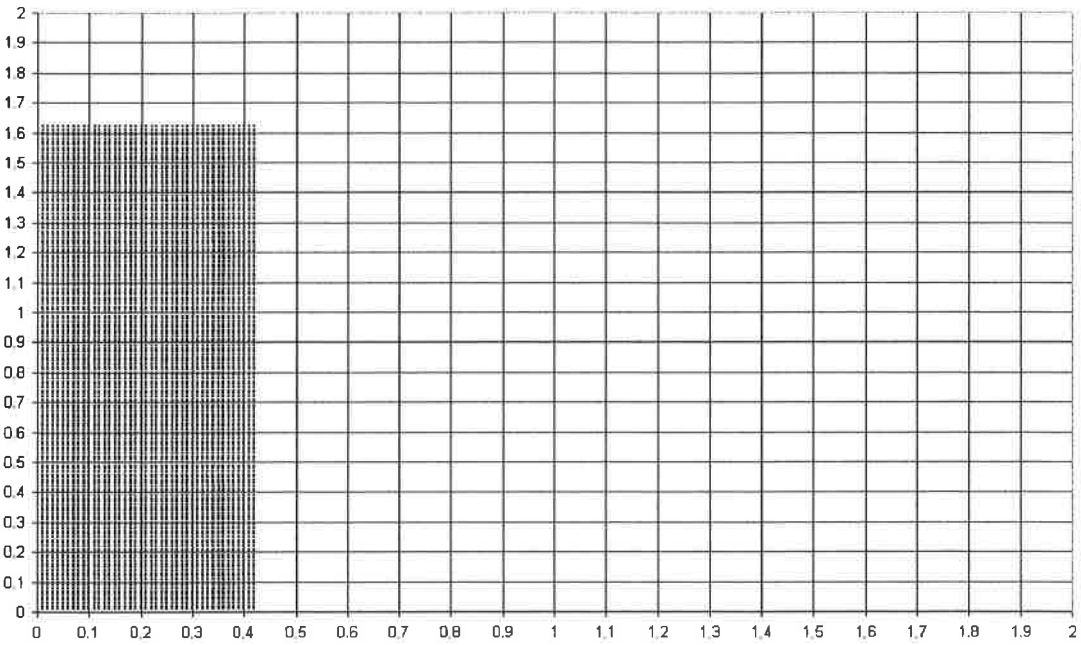


Fig 4.8 The initial configuration of the fluid at time 0.0s

Fig 4.9-4.15 show sample time frames at 0.2, 0.4, 0.6, 0.8, 1.0, 1.2 and 1.4 respectively. Again they clearly show the column of falling fluid accelerating as time progresses. The model's timestep has been reduced to 0.0001, which keeps the computation stable but increases the computational effort severely.

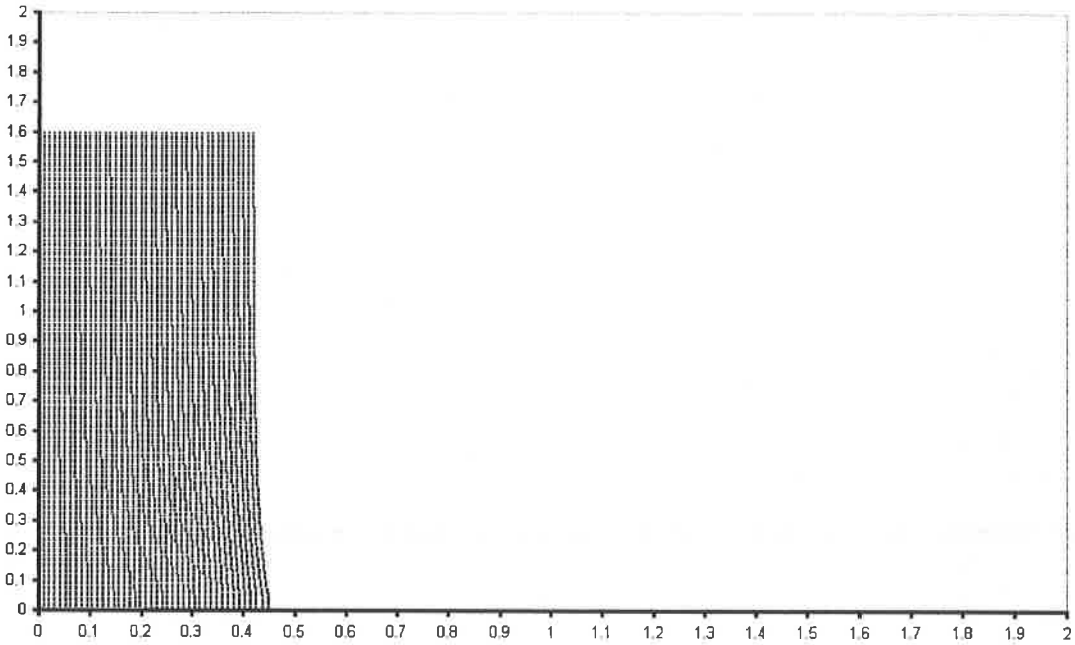


Fig 4.9 The evolution of the fluid at time 0.2

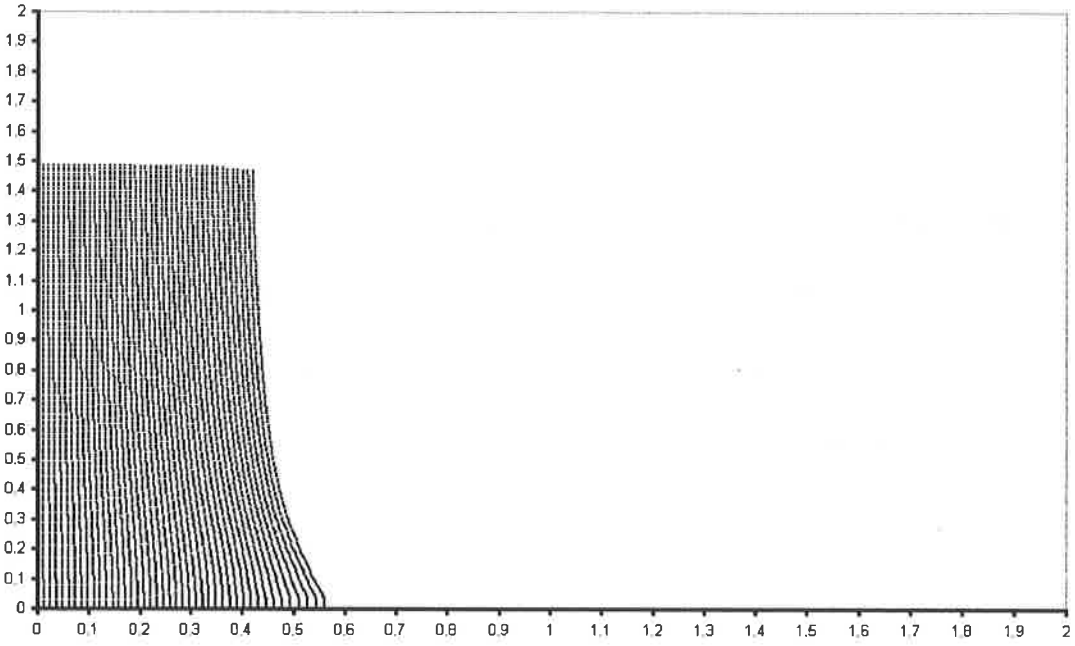


Fig 4.10 The evolution of the fluid at time 0.4

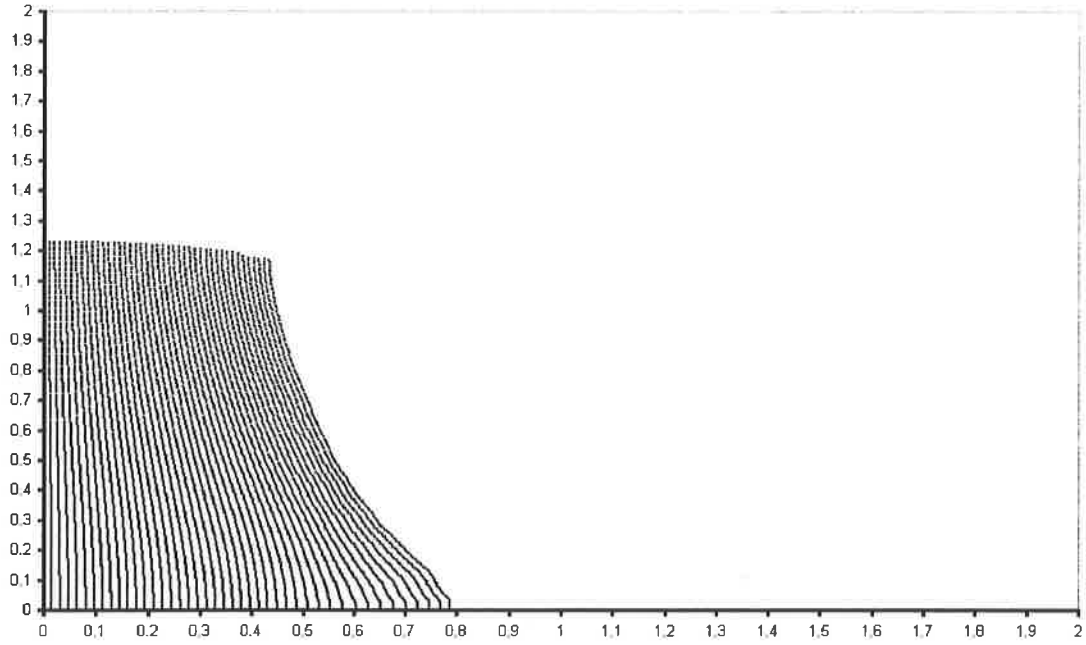


Fig 4.11 The evolution of the fluid at time 0.6

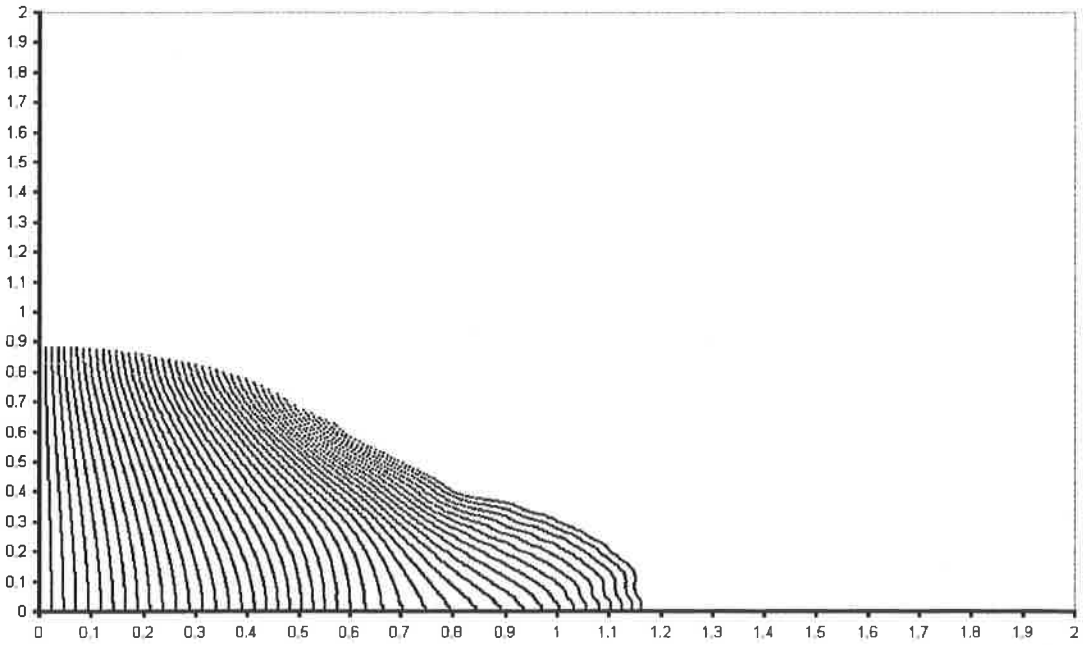


Fig 4.12 The evolution of the fluid at time 0.8

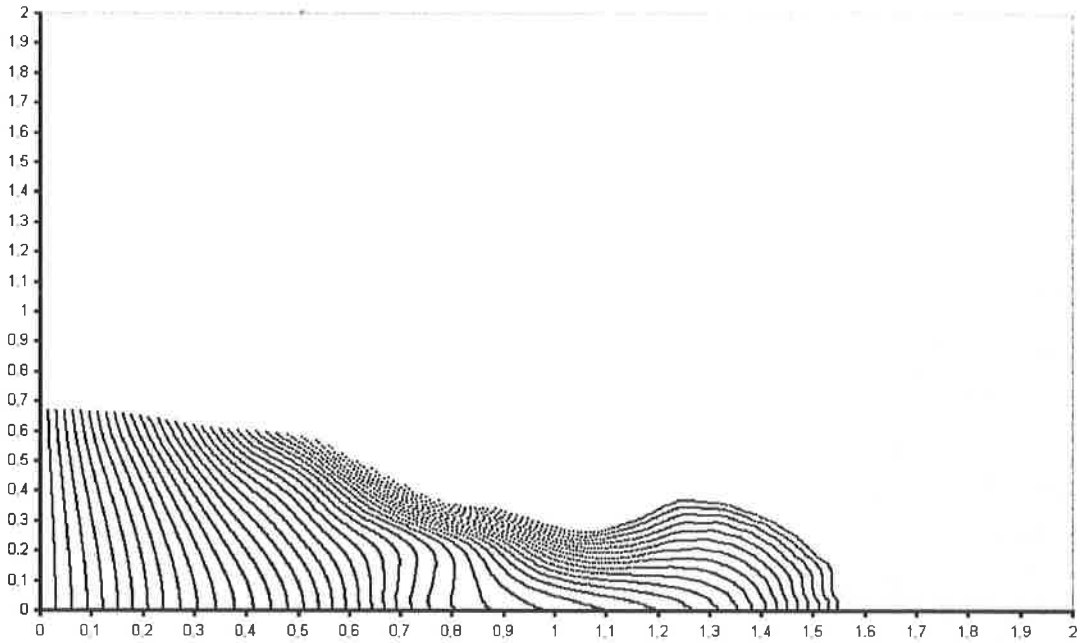


Fig 4.13 The evolution of the fluid at time 1.0

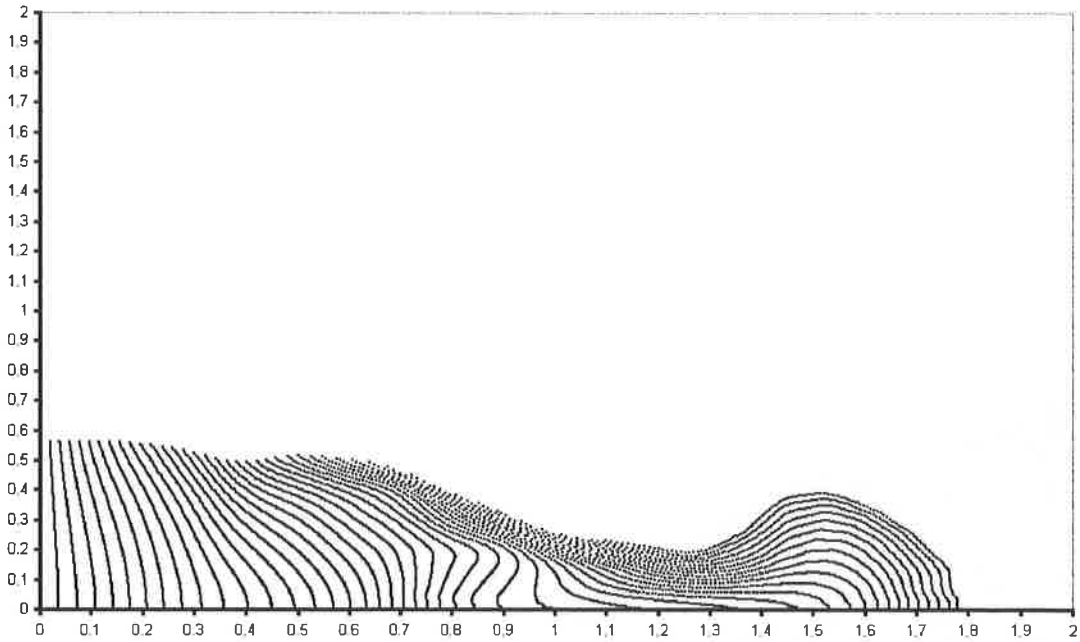


Fig 4.14 The evolution of the fluid at time 1.2

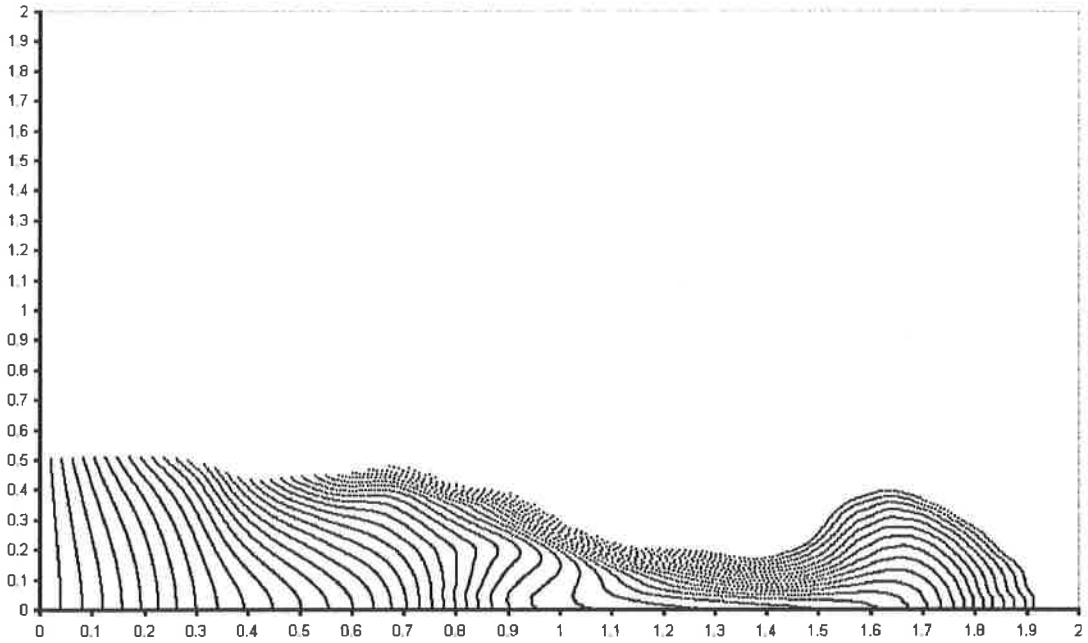


Fig 4.15 The evolution of the fluid at time 1.4

4.3 Fluid Column Collapse including reflected wave with Marker Particle Spacing 0.01 and 0.005 units

Fig 4.8 shows the initial configuration of the fluid at time 0.0s, the massless marker particles are spaced at 0.01 units for $0 < x < 0.35$ and are spaced at 0.005 units for $0.35 < x < 0.42$. This refinement is an attempt to capture the rapidly evolving interface due to the reflected wave.

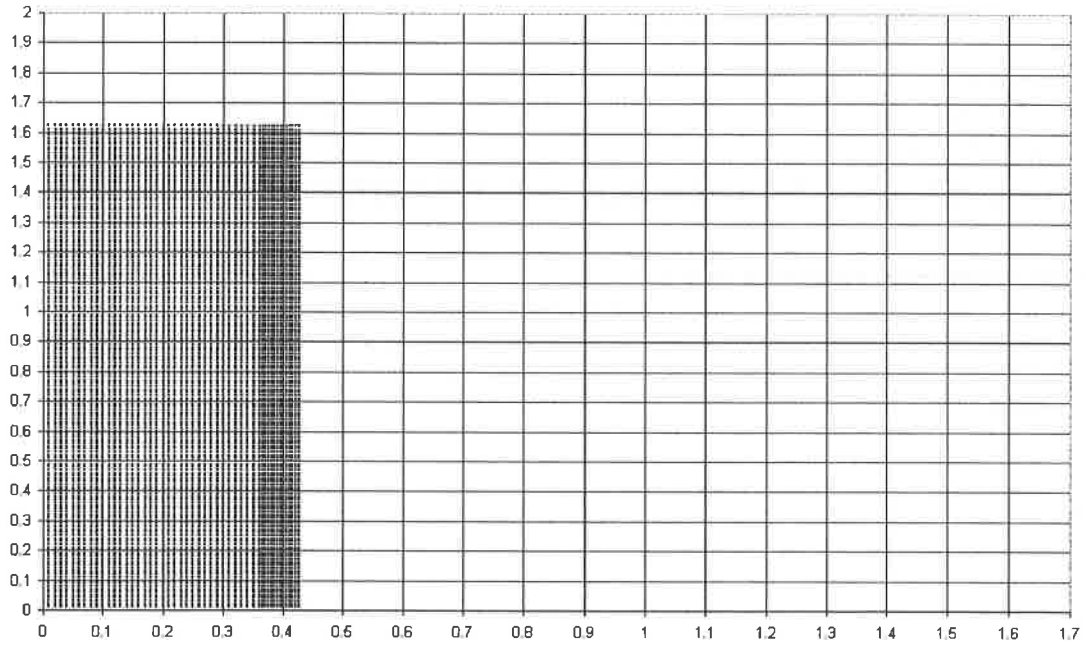


Fig 4.16 The initial configuration of the fluid at time 0.0s

Fig 4.17-4.22 show sample time frames at 0.4, 0.8, 1.0, 1.2, 1.4 and 1.6 respectively. With this model computation there are many errors in the final output due to factors such as the grid being too coarse, not enough marker particles being used and instability problems. The most significant error is that the grid is too coarse, this means that when there is folding in the flow, this will be captured for a certain amount of time before it assumes it has merged with the flow, due to the flagging scheme denoting it as a full cell and applying the relevant difference equations. Despite using adaptive timestepping the model initially displayed numerical instability in the later timesteps, which was due to computer rounding error in solving the discretised

Poisson equation for the velocity potential. It was found that the Poisson equation must be solved more accurately with the residual driven to less than 10^{-8} to maintain the numerical stability of the model and correct this problem.

The refinement in the grid works well to capture the interface effectively. It must be remembered, however, that an increase in the amount of marker particles used will increase the computational effort. Hence a computationally efficient method would be to monitor the velocity field and add or remove marker particles according to the evolution of the flow. This would also eliminate the problem of false voiding in the flow due to an insufficient amount of marker particles present.

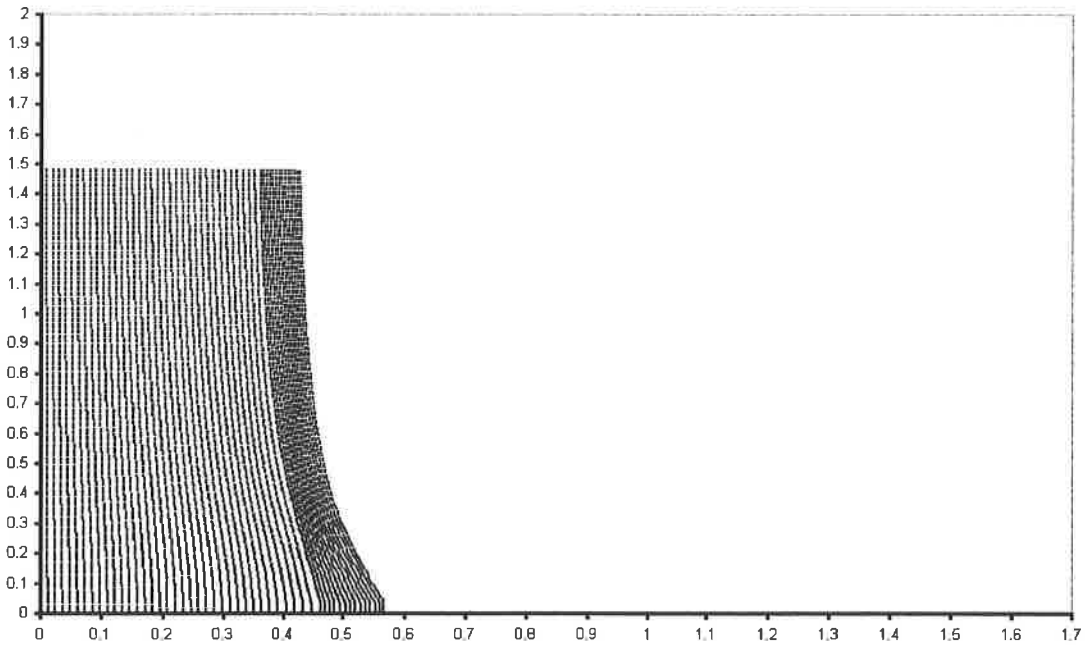


Fig 4.17 The evolution of the fluid at time 0.4

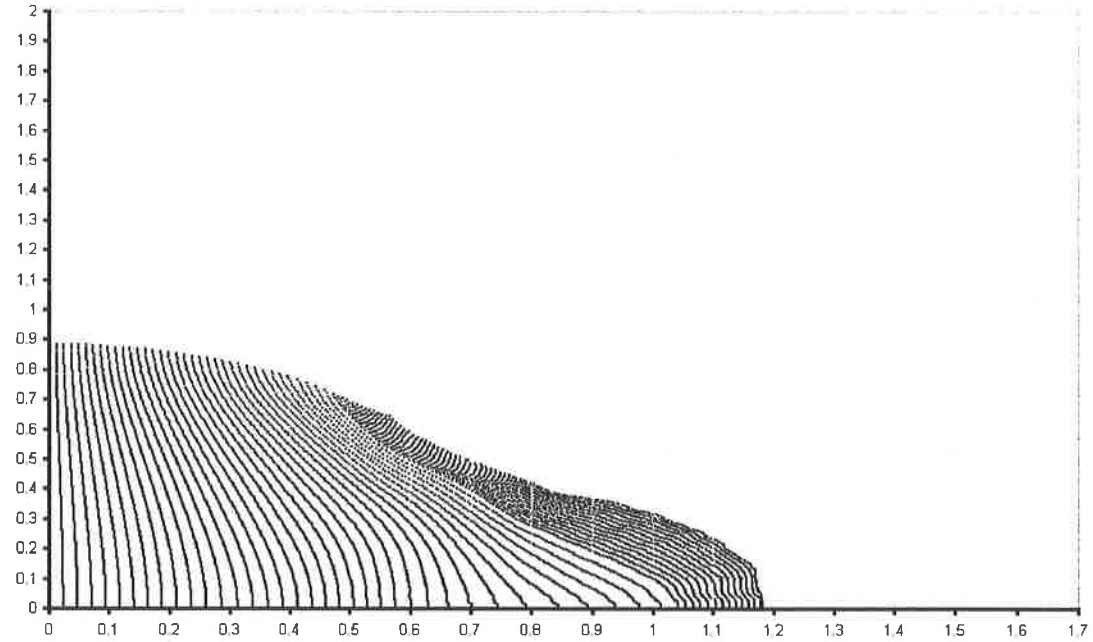


Fig 4.18 The evolution of the fluid at time 0.8

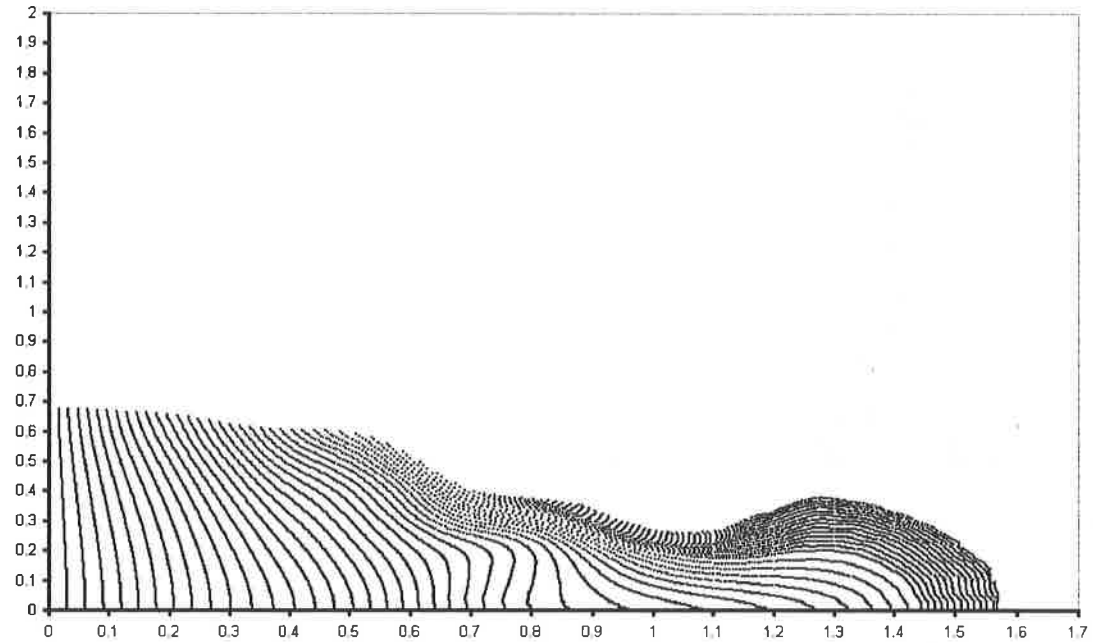


Fig 4.19 The evolution of the fluid at time 1.0

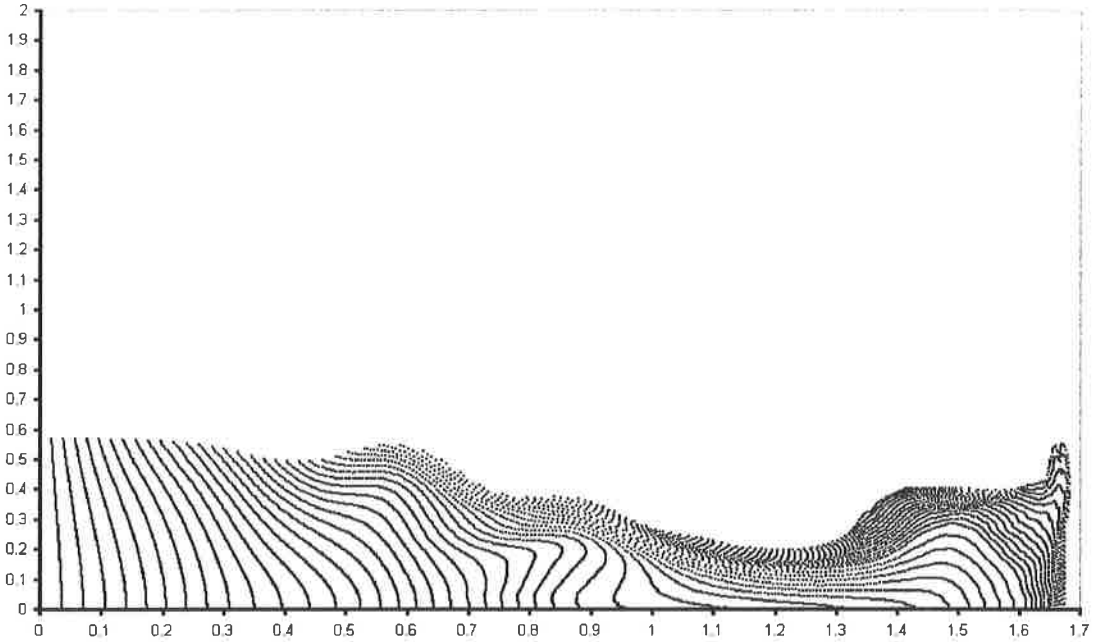


Fig 4.20 The evolution of the fluid at time 1.2

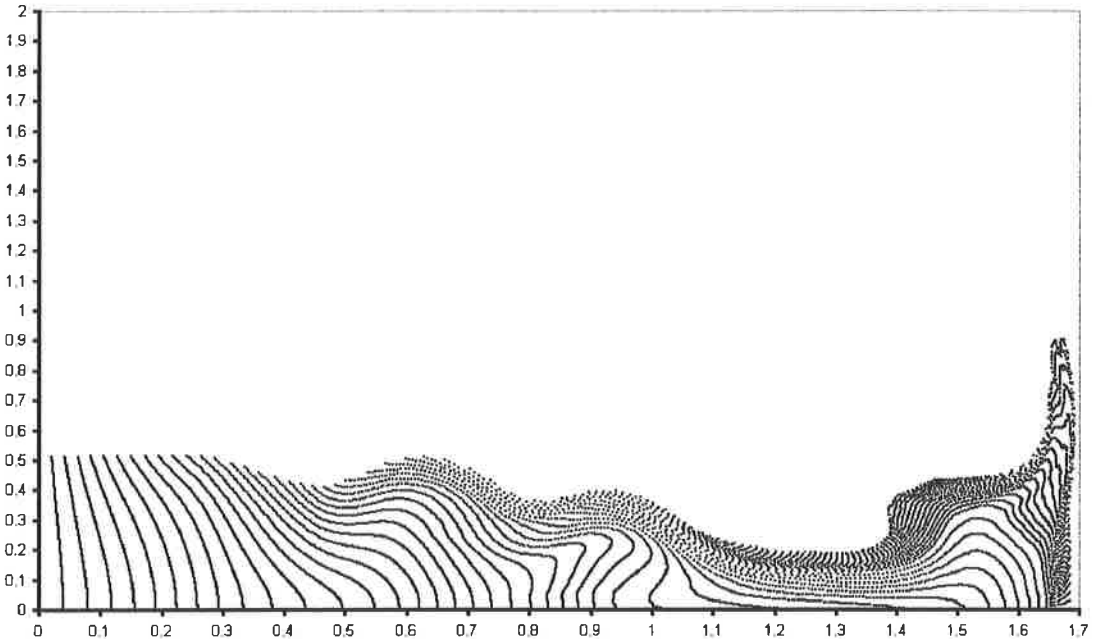


Fig 4.21 The evolution of the fluid at time 1.4

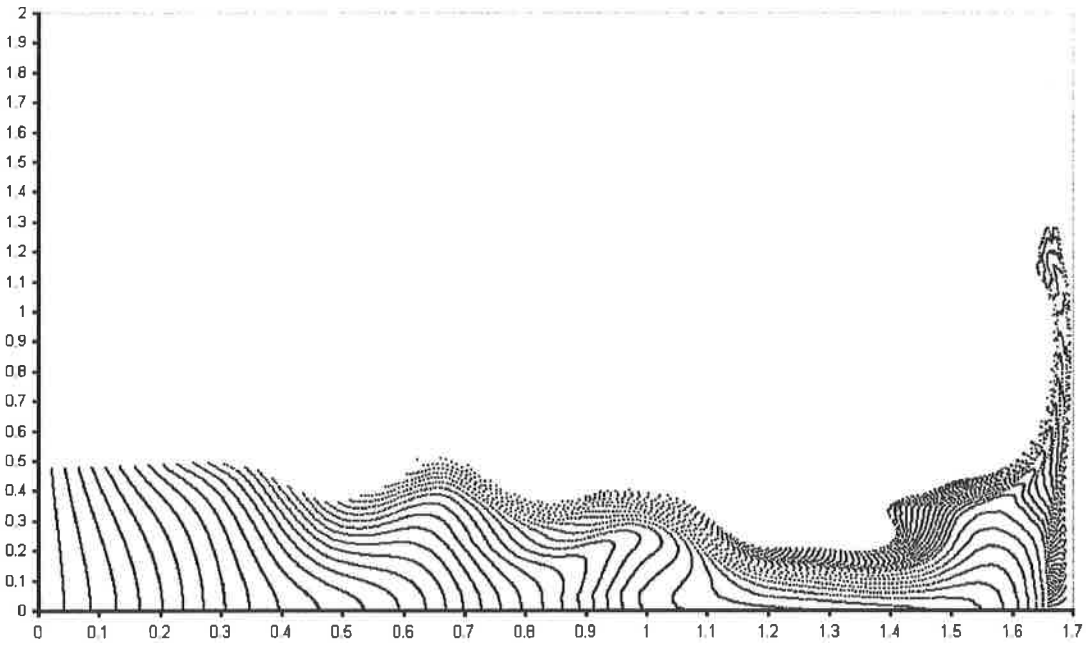


Fig 4.22 The evolution of the fluid at time 1.6

5.0 Suggested Improvements

The problem of the dam break over a dry bed has been solved in a very elementary manner. There is much that can be improved in the model, especially in the area of the free surface stress condition. It would also be interesting to test the code on other problems such as viscous bore modelling.

5.1 Accurate representation of the free surface boundary condition

Firstly recall that our free surface stress conditions are

$$\phi - \frac{2}{\mathrm{Re}}\left[n_x n_x \frac{\partial u}{\partial x} + n_x n_y \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + n_y n_y \frac{\partial v}{\partial y} \right] = 0 \qquad (5.1)$$

$$\left[2 n_x m_x \frac{\partial u}{\partial x} + \left(n_x m_y + n_y m_x \right) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + 2 n_y m_y \frac{\partial v}{\partial y} \right] = 0 \qquad (5.2)$$

When problems like this were first solved, the pressure was set to zero at all surface cells. Later methods were modified to include the normal viscous stress in the surface cell pressure, but the pressure was still specified at the cell centre, regardless of the location of the surface within the cell.

However, in this method we use a marker and cell technique to represent the free surface. If we used a volume method such as that of Youngs' then this would include the calculation of the average normal of the free surface by application of the divergence theorem (Youngs, 1982). We can use this additional information to represent 5.1, more accurately. Whereas previously we assumed that either n_x or n_y is small, we no longer have to make this assumption since we hold the value of the normal n, where $n = (n_x, n_y)$. Now, still assuming that a surface falls into categories 1,2 or 3, then we can calculate the average normal for the cell and then apply it to Eq 5.1 to gain an accurate representation of the free surface stress condition.

Furthermore, following Nichols and Hirt, we can apply the pressure directly to the free surface by means of linear interpolation.

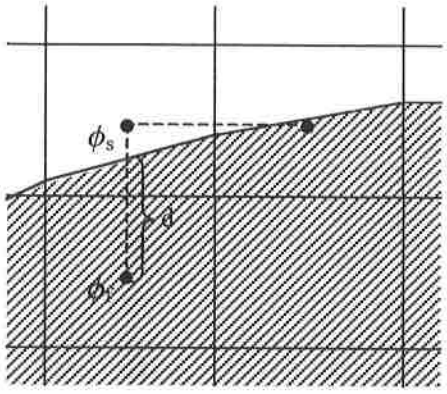


Fig 5.1 A typical cell interpolation

The pressure at the cell centre is specified as the linear interpolation between an adjacent full cell pressure and the required pressure at the fluid surface. The interpolation formula is

$$\phi_s = \eta(\phi_a + \phi_{ns}) + (1-\eta)\phi_f \tag{5.3}$$

where ϕ_{ns} is the surface pressure from the normal stress condition Eq. 5.1, ϕ_a is any additional pressure applied to the free surface, ϕ_f is the current value of pressure in the neighbouring cell, and η is the interpolation factor which can be determined following Fig 5.1 as

$$\eta = \frac{\delta y}{d} \tag{5.4}$$

A complete description of this method can be found in Nichols and Hirt, 1971.

Again, following this procedure the tangential stress condition Eq. 5.2 can now be fully represented without any assumptions, since we have information about the surface normal and configuration.

5.2 Additional Test Problems

It would also be interesting to test the model further using the no-slip condition at the domain boundary. At present for ease of implementation, the free slip condition is applied to domain boundaries, but for our problem, this is not physically correct and the correct domain boundary condition is that of no slip. This would induce a boundary layer to form.

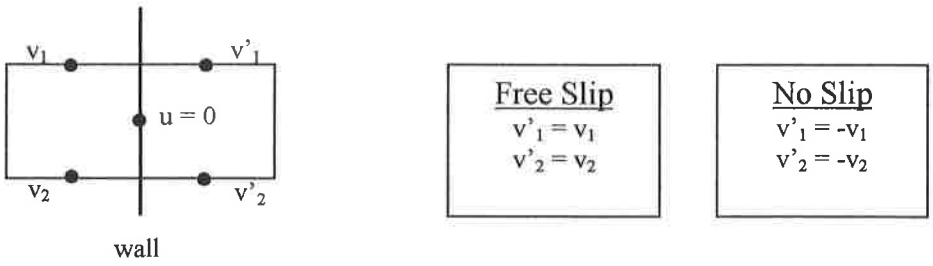


Fig 5.2 The configuration of the free slip and no slip conditions

The sole difference between the free slip and the no slip slip condition as shown by Fig 5.2 is that for no slip $v'_i = -v_i$. However to model this condition properly a very fine and geometrically progressive grid must be used around the area of the boundary layer and this would have been beyond the scope of this study. Further attention needs to be devoted to implementation of the no-slip condition.

5.2.1 Dam Break with an Obstacle

This test problem would involve all types of interfaces of the flow, resulting in empty cells at the top, bottom, left and right of the flow. It would also be ideal to model detachments to the flow and merging fluids. The initial set up is illustrated by Fig 5.3, which is identical to our initial set up for the dam break problem but with the inclusion of an obstacle as shown.

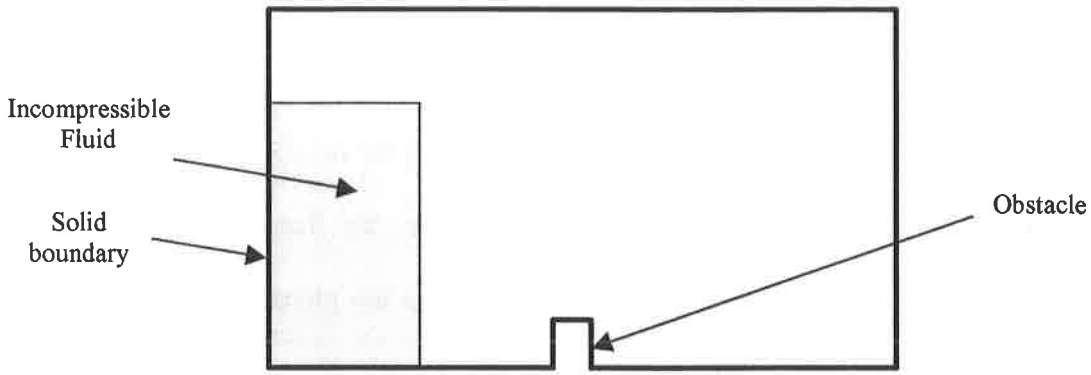


Fig 5.3 The initial set-up of a dam break problem on a dry bed with an obstacle in the flow.

This problem would require a well refined grid. This would mean a geometrically progressive grid, well refined at the boundary of the domain, and sufficiently fine to accommodate high curvature which would be present in a problem such as this.

6.0 Conclusions

The main objective of this study was to solve the incompressible Navier-Stokes equations using a finite difference representation in a Eulerian framework. This has been achieved using a pressure correction philosophy and a Marker and Cell method of tracking the fluid interface. The problem studied was that of a dam break problem on a dry bed. The results of the computation showed that the problem could only be solved successfully with a small spacing of 0.01 between the marker particles. The full algorithm for the solution can be stated fully as :

1. Setup the initial configuration of the fluid by use of massless marker particles,
2. Determine the flagging of the domain for the timestep,
3. Let $\tilde{\phi}(x,t_o)$ be an arbitrary pressure field which satisfies the correct boundary conditions at the free surface,
4. Calculate the intermediate velocity field $\tilde{u}(x,t)$, from

$$\frac{\partial \tilde{u}}{\partial t} = \left[-\frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} - \frac{\partial \tilde{\phi}}{\partial x} + \frac{1}{\mathrm{Re}} \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + \left(1/Fr^2 \right) g_x \right]_{t=t_o} \qquad (3.34)$$

$$\frac{\partial \tilde{v}}{\partial t} = \left[-\frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} - \frac{\partial \tilde{\phi}}{\partial y} - \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + \left(1/Fr^2 \right) g_y \right]_{t=t_o} \qquad (3.35)$$

5. If the timestep used $\delta t > MIN\{\delta t_{visc}, \delta t_u, \delta t_v\} \cdot A$ then correct the timestep and goto 4,
6. Solve the Poisson equation,

$$\nabla^2 \Psi = \nabla \cdot \tilde{u}(x,t) \qquad (3.36)$$

7. Compute the velocity field,

$$u(x,t) = \tilde{u}(x,t) - \nabla\Psi \tag{3.37}$$

8. Compute the pressure field,

$$\phi = \tilde{\phi} + \frac{\Psi}{\delta t} \tag{3.38}$$

9. Update the marker particle position position by Euler's method using

$$x_p^{n+1} = x_p^n + u_p \delta t \tag{3.60a}$$

$$y_p^{n+1} = y_p^n + v_p \delta t \tag{3.60b}$$

10. Update the timestep so t = t + δt and goto 2

The model used an adaptive time stepping procedure in an attempt satisfy the CFL condition, maintaining numerical stability and also reducing computational effort. However, it was found that even when using this adaptive timestepping procedure the model could still demonstrate instability. This may be due to solving the Poisson equation for the potential function with insufficient accuracy. This study has also included a review of current interface capture methods and suggested further improvements to the model.

References

[1] F. Tome and S. McKee, 'GENSMAC : A Computational Marker and Cell Method for Free Surface Flows in General Domains' J Comput. Phys, Vol 110, 171-186, 1994

[2] C. W.Hirt and B.D. Nichols, 'Volume of Fluid Method for the Dynamics of Free Boundaries', J Comput. Phys, Vol 39, 201-225, 1981

[3] O. Ubbink, 'Numerical Prediction of two Fluid Systems with Sharp Interfaces', PhD Thesis, Imperial College, 1997

[4] J.D. Anderson, Computational Fluid Dynamics, The Basics with Applications – International Edition, McGraw-Hill Publishing, 1995

[5] B.D. Nichols and C.W. Hirt, 'Improved Free Surface Boundary Conditions for Numerical Incompressible Flow Calculations J Comput. Phys, Vol 8, 434-448, 1971

[6] N. Ashgriz and J.Y. Poo, 'FLAIR : Flux-Line Segment Model for Advection and Interface Reconstruction', J Comput. Phys, Vol 93, 449-468, 1991

[7] D. Login, 'Lecture Notes : Advanced Boundary Value Problems', Department of Mathematics, University of Reading, 2000

[8] F.H. Harlow and J.E. Welch, 'Numerical Calculation of Time Dependent Viscous Incmpressible Flow of Fluid with Free Surface', Phys. Fluids, Vol 8(12), 2182-2189,1965

[9] R. Giddings, 'HELMIT – A New Interface Reconstruction Algorithm', Presentation Slides for Atomic Weapons Estabilishment, International Computational Fluid Dynamics Seminar, Oxford, 1999

[10] M. Rudman, 'Volume Tracking Methods for Interfacial Flow Calculations', International Journal for Numerical Methods in Fluids, Vol 24, 671-691, 1997

[11] B.J. Daly, 'A Technique for including surface tension effects in hydrodynamic calculations.', J Comput. Phys, Vol 4, 97-117,1969

[12] M.S. Darwish, 'A New High Resolution Scheme Based on the Normalised Variable Formulation', Numerical Heat Transfer, Part B, Vol 24, 353-371, 1993

[13] C.W. Hirt and B.D. Nichols, 'Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries', J Comput. Phys, Vol 39, 201-225, 1981

[14] C.W. Hirt and B.D. Nichols, 'Calculating Three Dimensional Free Surface Flows in the Vicinity of submerged and Exposed Strucures', J Comput. Phys, Vol 12, 234-246, 1973

[15] S. Osher and J.A. Sethian, 'Fronts Propagating with Curvature Dependent Speed : Algorithms Based on the Hamilton-Jacobi Formulations', J Comput. Phys, Vol 79, 12-49, 1988

[16] D.L. Youngs, 'Time Dependent Multi Material Flow with Large Fluid Distortion', In : K.W. Morton and M.J. Baines (eds) Numerical Methods for Fluid Dynamics, London, Academic Press, 273-285, 1982

[17] W.F. Noh and P. Woodward, 'SLIC : Simple Line Interface Calculations', Lecture Notes in Physics, Vol 59, 330-340, 1976

Acknowledgements

I would initially like to thank my project supervisors Prof. M.J. Baines and Dr P. Glaister for their guidance in completing this project and many valuable suggestions.

I would also like to acknowledge Mr R Giddings for his help in my understanding of current Volume of Fluid methods and particularly his extension using the HELMIT algorithm.

I am also grateful to my lecturers and colleagues in the Department of Mathematics for many productive discussions, and also Mr. D. Login of the Oxford University Computer Laboratory for his help with the solution of large sparse systems and sparse storage schemes.

This could all only be achieved, however, with the support and encouragement of my family, and the patience of Polly Few.

Finally I would like to acknowledge the effort of Mrs S. Davis in her organisation of the course, which is indeed a credit to her.

Appendix A

Linear Algebra Algorithms

CSR Matrix by Vector Algorithm

Using CSR storage, an effective algorithm to multiply a matrix A by a vector x would be as follows

Let $\underline{Y} = A\underline{x}$

for $k = 1,2, \ldots, n$

$$\alpha_1 = RPt(k)$$

$$\alpha_2 = RPt(k+1)$$

$$Y(k) = A(\alpha_1 : \alpha_2) \cdot X(Col(\alpha_1 : \alpha_2))$$

end

Each loop of the algorithm computes a different component of the vector. This is an advantage since it allows any component of the vector to be computed independently.

The Conjugate Gradient Algorithm

The algorithm for the conjugate gradient method as given by Login [7] is as follows,

$x^o \in \Re^n$; *compute* $d^o = r^o = b - Ax^o$

for $k = 1, 2, \ldots, n$ *compute*

if $k = 1$

$$d^1 = r^o$$

else

$$\beta_k = -\frac{d^{k-1} \cdot Ar^{k-1}}{d^{k-1} \cdot Ad^{k-1}}$$

$$d^k = r^{k-1} + \beta_k d^{k-1}$$

end if

$$\alpha_k = \frac{d^k \cdot r^{k-1}}{d^k \cdot Ad^k}$$

$$x^k = x^{k-1} + \alpha_k d^k$$

$$r^k = b - Ax^k$$

end

This is an effective robust algorithm for solving the linear system, $A\underline{x} = \underline{b}$.

Appendix B

HELMIT – Hinged Line Method of Interface Tracking

The disadvantage of the schemes discussed in Section 2.2.2 is the fact that they all have discontinuities in the representation of the fluid surface, as depicted in Fig B1

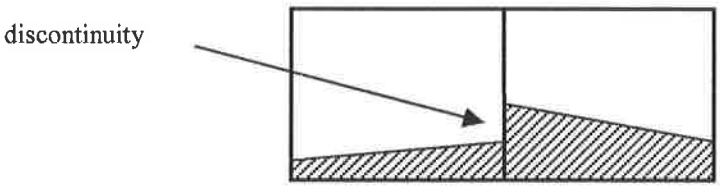


Fig B1 – A discontinuity in the interface representation

HELMIT described fully in Giddings [9] goes some way to correcting this problem. The algorithm involves three main steps.

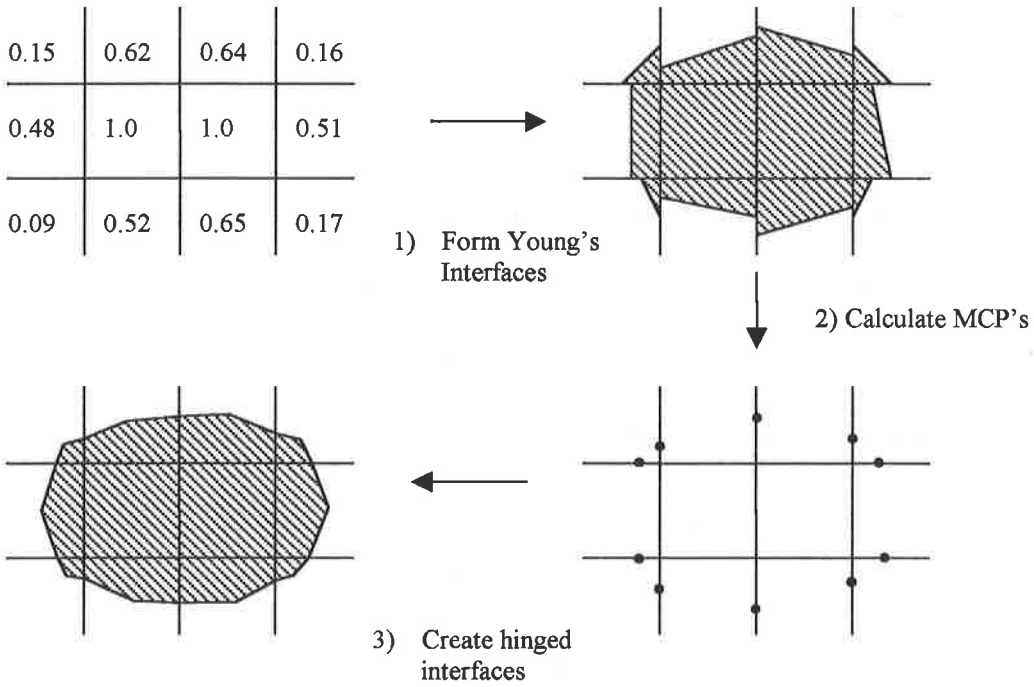


Fig B2 – The sequence of steps in obtaining the hinged interfaces

Step 1 : Use a Technique such as Young's method to form initial interfaces.

Step 2 : Where cell interfaces do not meet, force them to go through a 'mesh crossing point' (MCP) which is halfway between the interfaces as shown by Fig B2

Step 3 : Once the MCP's have been placed. The points are joined with a straight line, then an isosceles triangle is used to add or remove volume to adjust the volume fraction to the correct amount as shown by Fig B3.

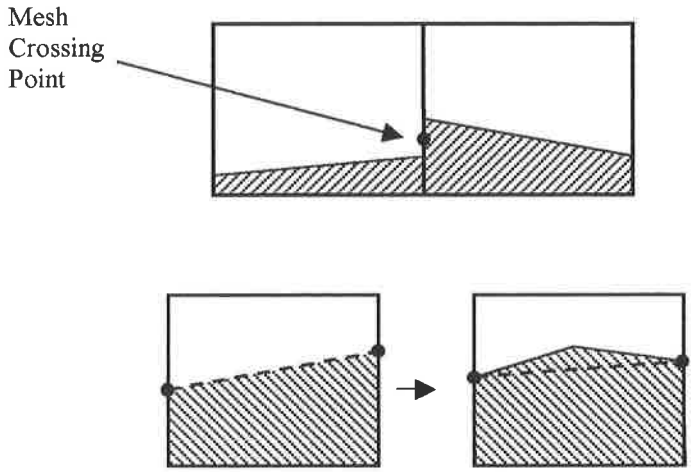


Fig B3 – The adjustment of the hinged surface to represent the volume fraction

Each interface now consists of the two lines which are hinged. This method is known as the Hinged Line Method of Interface Tracking or HELMIT.

3.3.1 Exceptional Cases for HELMIT

The first exceptional case is when the volume fraction is adjusted using the isosceles triangles, when it is possible that the apex lies outside of the cell. When this happens the apex is retracted along the perpendicular bisector of the base until it lies on the cell boundary. The MCP's are then moved to correct the area.

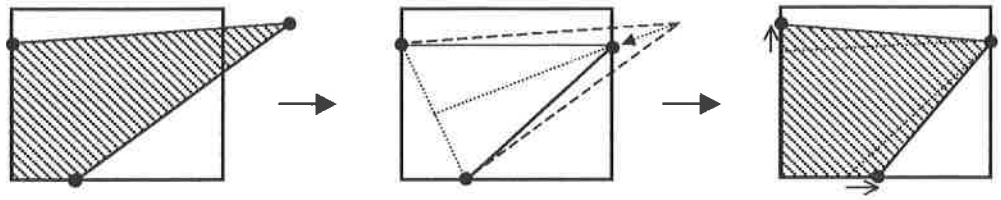


Fig B4 – An exceptional case of the adjustment moving outside of the cell boundary

The second exceptional case is that it is not always possible to find a suitable pair of interfaces with which to create the MCP; an interface may have no partner or indeed many.

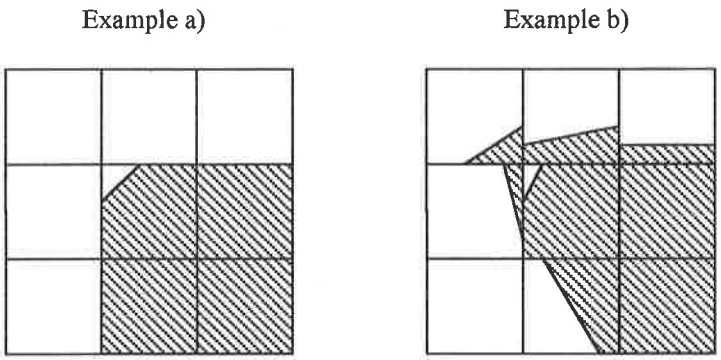


Fig B5 – An exceptional case of no suitable pair of interfaces available to create MCP

For example a) this suggests that the interfaces themselves are not as important as the body of fluid they enclose. Example b) shows a problem in defining the MCP for the top centre cell and this is due to the interface in the centre cell. HELMIT deals with this by theoretically filling in the centre cell and thus MCPs can be uniquely identified.

3.3.2 Half HELMIT and Double HELMIT

When HELMIT is implemented in a code it must have available to it all Youngs' interfaces, which means all interfaces must be stored since they may be used up to 5

times per interface. Half HELMIT avoids this by reusing information calculated in the original Youngs' step. Youngs' takes each bordering cell in turn and fits a straight line through that cell and the centre cell, satisfying both volume fractions. This is known as two square fittings.

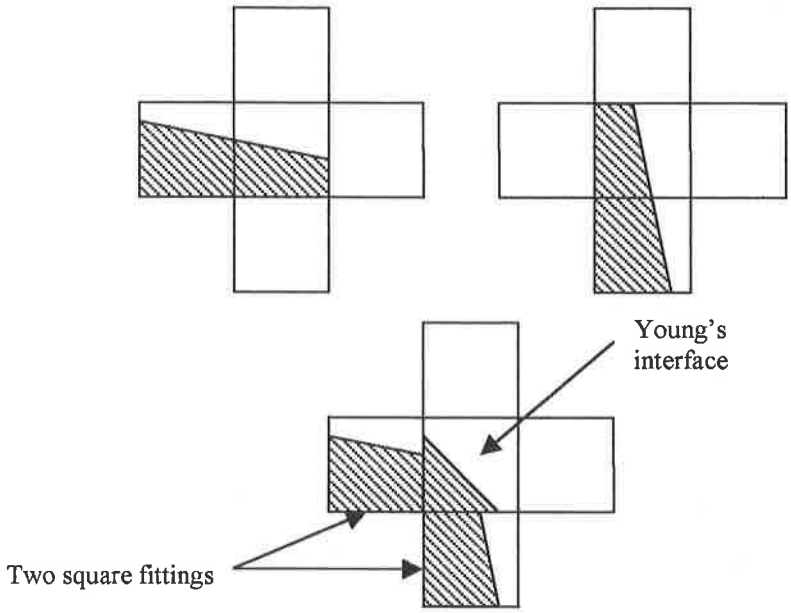


Fig B6 – The relation between two square fitting and the Youngs' interface

Half HELMIT stores and uses only the length of the common border between the two cells containing material and discards everything else. MCP's used to be placed between two interfaces to make sure they joined up. However now there is only one proper interface, in the centre cell and therefore placing an MCP using two Youngs interfaces is no longer possible. It is therefore decided that the MCP should be placed a fixed fraction ($0 \leq \lambda \leq 1$) of the distance from the Youngs interface to the two square fitting. A value of $\lambda = 0.5$ is often a reasonable choice.

Double HELMIT puts MCPs halfway between two Half HELMIT interfaces, which is computationally more expensive but a more accurate representation of the surface.