INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **54**:1033–1053
Published online 21 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1409

The development of a Cartesian cut cell method for incompressible viscous flows

Feng Gao[1,*,†], D. M. Ingram[2], D. M. Causon[1] and C. G. Mingham[1]

[1] *Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University,*
Chester Street, Manchester M1 5GD, U.K.
[2] *Institute for Energy Systems, The University of Edinburgh, The King's Buildings, Mayfield Road,*
Edinburgh EH9 3JL, U.K.

SUMMARY

This paper describes the extension of the Cartesian cut cell method to applications involving unsteady incompressible viscous fluid flow. The underlying scheme is based on the solution of the full Navier–Stokes equations for a variable density fluid system using the artificial compressibility technique together with a Jameson-type dual time iteration. The computational domain encompasses two fluid regions and the interface between them is treated as a contact discontinuity in the density field, thereby eliminating the need for special free surface tracking procedures. The Cartesian cut cell technique is used for fitting the complex geometry of solid boundaries across a stationary background Cartesian grid which is located inside the computational domain. A time accurate solution is achieved by using an implicit dual-time iteration technique based on a slope-limited, high-order, Godunov-type scheme for the inviscid fluxes, while the viscous fluxes are estimated using central differencing. Validation of the new technique is by modelling the unsteady Couette flow and the Rayleigh–Taylor instability problems. Finally, a test case for wave run-up and overtopping over an impermeable sea dike is performed. Copyright © 2006 John Wiley & Sons, Ltd.

Received 5 June 2006; Revised 20 October 2006; Accepted 25 October 2006

KEY WORDS: Cartesian cut cell; incompressible; viscous flow; free surface

1. INTRODUCTION

Motions of fluids with a free surface are important phenomena in many fields of civil and coastal engineering and this has led to the development of a range of numerical modelling techniques.

[*] Correspondence to: Feng Gao, Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, U.K.
[†] E-mail: f.gao@mmu.ac.uk

Contract/grant sponsor: EPSRC; contract/grant number: GR/T18622/01
Contract/grant sponsor: Scottish Funding Council

Accurate simulation of such flows is a difficult problem since the location of the free surface is not only driven by the gross motions of the fluid but drives that fluid motion. To be useful, the selected technique must be able to cope with a free surface which becomes highly distorted, multiply valued, and which undergoes gross topological changes including break-up and recombination. The successful approaches for handling free surface problems can be broadly grouped into two categories: surface fitting methods and surface tracking methods.

Surface fitting methods solve the flow in the fluid region only and the free surface is treated as a moving boundary of the computational domain, which satisfies the kinematic boundary conditions. In this technique, free surface locations can be recognized precisely. Several researchers [1–3] have applied this method to a variety of free surface flow problems. However, this technique requires frequent repositioning of computational grid points, thus, numerical errors and lower numerical stability may occur. Furthermore, when the free surface becomes highly distorted, during wave breaking, for example, the numerical simulations become difficult to perform.

Surface tracking methods simulate both fluid regions on a fixed grid system, with the free surface being identified by a marker function such as the marker particles in the marker and cell (MAC) method [4] or the volume fraction function in the volume of fluid (VOF) method [5–7]. In these methods, a transport equation for the marker function is solved at each time step and the shape of the free surface can then be reconstructed from the distribution of the marker fraction. This method can define sharp interfaces and is robust. However, the tracking and reconstruction of free surfaces remains complicated and difficult, especially in three dimensions [8].

Several alternative methods have appeared in recent years which address some of the shortcomings of the previous surface fitting and surface tracking schemes. Unverdi and Tryggvason [9] developed a front-tracking scheme which solved for the liquid and gas flow fields on a stationary grid and represented the free surface using a separate unstructured grid. Sussman *et al.* [10] presented an approach based on the level set method, which used the zero level set of a smooth function to identify the free surface. Kelecy and Pletcher [11] presented a free surface capturing technique which computed free surface flows within closed containers, but the numerical smearing of the free surface was quite severe. Pan and Chang [12] reduced the smearing by introducing a slope modification method. More recently, Qian *et al.* [13, 14] extended this technique to inviscid flow fluid calculations in a Cartesian cut cell framework and successfully simulated free surface flows involving the gross motion of solid objects. This approach views the free surface as a discontinuity in the density field and the material interface is automatically captured as part of the evolving numerical solution, along with other flow variables such pressure and velocity, by enforcement of a conservation law.

This paper gives brief details of an extended version of the inviscid, surface capturing solver discussed by Qian *et al.* [13, 14]. The present code solves the incompressible Navier–Stokes equations and utilizes the Cartesian cut cell technique in order to generate boundary conforming mesh representations of any solid bodies inside the computational domain. This has the advantage that complex geometric or moving bodies in the flow domain can be easily represented [15–17].

To demonstrate the present method, results are presented for unsteady Couette flow, Rayleigh–Taylor instability, and the wave run-up and subsequent overtopping over an impermeable sea dike.

DOI: 10.1002/fld

2. GOVERNING EQUATIONS

For incompressible, unsteady, viscous flows, the two-dimensional Navier–Stokes equations with a variable density field can be modified by the artificial compressibility method and written as the integral form

$$\frac{\partial}{\partial \mathbf{t}}\iint_{\Omega} \mathbf{Q}\,\mathbf{d}\Omega + \oint_{S} \mathbf{F}\cdot n\,\mathbf{d}s = \iint_{\Omega} \mathbf{B}\,\mathbf{d}\Omega \tag{1}$$

where Ω is the domain of interest; S is the boundary surrounding Ω, $\mathbf{n}$ is the unit normal to S in the outward direction. $\mathbf{Q}$ is the vector of conserved variables. $\mathbf{F}$ is the vector of flux function through S and $\mathbf{B}$ is the body forcing function. These equations in order are conservation laws for the mass (density) and two components of momentum, together with an incompressibility constraint, which adds a time derivative of pressure to the continuity equation. By using the artificial compressibility method and assuming the only body force is gravity, **Q, F** and **B** are

$$\begin{aligned}
\mathbf{Q} &= [\rho,\ \ \rho u,\ \ \rho v,\ \ p/\beta]^{\mathrm{T}}\\
\mathbf{F} &= \mathbf{F}^{\mathrm{I}} - \mathbf{F}^{\mathrm{V}} = (\mathbf{f}^{\mathrm{I}} - \mathbf{f}^{\mathrm{V}})\mathbf{n_x} + (\mathbf{h}^{\mathrm{I}} - \mathbf{h}^{\mathrm{V}})\mathbf{n_y}\\
\mathbf{B} &= [0,\ \ 0,\ \ -\rho\mathbf{g},\ \ 0]^{\mathrm{T}}\\
\mathbf{f}^{\mathrm{I}} &= [\rho u,\ \ \rho u^2 + p,\ \ \rho uv,\ \ u]^{\mathrm{T}}\\
\mathbf{h}^{\mathrm{I}} &= [\rho v,\ \ \rho uv,\ \ \rho v^2 + p,\ \ v]^{\mathrm{T}}\\
\mathbf{f}^{\mathrm{V}} &= \left[0,\ \ \mu\frac{\partial u}{\partial x},\ \ \mu\frac{\partial v}{\partial x},\ \ 0\right]^{\mathrm{T}}\\
\mathbf{h}^{\mathrm{V}} &= \left[0,\ \ \mu\frac{\partial u}{\partial y},\ \ \mu\frac{\partial v}{\partial y},\ \ 0\right]^{\mathrm{T}}
\end{aligned} \tag{2}$$

where the superscripts I and V denote the inviscid and viscous flux, respectively, $\mathbf{n_x}$ and $\mathbf{n_y}$ are the unit vectors along x and y co-ordinate directions, ρ is the density, p is the hydrodynamic pressure, u and v are the velocity components, β is the coefficient of artificial compressibility, $\mathbf{g}$ is the gravitational acceleration and μ is the dynamic viscosity coefficient.

The addition of the pseudo-time derivative term in the continuity equation directly couples the pressure and velocity at the same time level, which make the governing equations hyperbolic in space–time [18]. The new system of equations may then be iterated in pseudo-time, τ, until a steady state is achieved and the divergence-free velocity field condition is satisfied. Several well-known and stable time-marching techniques can be employed, which include a finite difference method [19, 20], an explicit multi-stage scheme [21], a flux-difference splitting scheme [22], a total variation diminishing (TVD) scheme [12] and a characteristics-based Godunov-type scheme [13]. In the steady state, the pseudo-time derivative term vanishes, satisfying the conservation of mass. For unsteady flow problems, however, a divergence-free velocity field must be attained at every time step. This can be achieved by using a dual-time stepping technique [21] and sub-iterating the equations in the pseudo-time domain to achieve a steady-state solution at each physical time step.

DOI: 10.1002/fld

3. BOUNDARY CONDITIONS

The conditions at the boundaries enclosing the computational domain must be specified to completely define the problem. In the present study for hydraulic flow problems, there are four kinds of boundaries and a special treatment for the free surface is not required. The exact treatment of each boundary depends on the number of inward and outward pointing characteristics at the boundary. Since the system of equations under consideration is hyperbolic, all the characteristic speeds (eigenvalues) are real. For a boundary aligned with the y-ordinate, the four relevant characteristic speeds are

$$\lambda = u - \sqrt{u^2 + \beta}, 0, u, u + \sqrt{u^2 + \beta} \tag{3}$$

(i) *Inflow boundary*: The two components of velocity are specified at this boundary, the pressure and density are extrapolated from the interior of the computational domain by assuming a zero gradient. This definition allows the desired waves to propagate into the computational domain through this boundary.

(ii) *Outflow boundary*: The pressure at this boundary is fixed and a zero gradient condition is applied to the velocity and density. This definition allows fluids to freely enter or leave the computational domain according to the local flow velocity and direction.

(iii) *Solid wall boundary*: At these boundaries, which are represented using Cartesian cut cells, the no-slip condition can be applied to the velocity. Adopting a mirror-type wall model, the normal and parallel components of velocity in the ghost cell, located behind the boundary, are equal in magnitude to those inside but with opposite directions. As a consequence, all convective flux components through the solid boundary will vanish as soon as the steady condition is reached. For the remaining condition it is assumed that the density has a zero normal gradient. For the pressure, if the wall is stationary, we have $\nabla p = [0, -\rho\mathbf{g}]$.

(iv) *Symmetry boundary*: These may be implemented either in a cut cell, to generate a slip wall, or in an uncut Cartesian cell. The implementation is similar to the no-slip condition (iii) except that the velocity in the ghost cell is determined so as to impose the symmetry condition, $\mathbf{q} \cdot \mathbf{n} = 0$, where $\mathbf{q} = (u, v)^{\mathrm{T}}$ and $\mathbf{n}$ is the wall normal vector.

4. CARTESIAN CUT CELL MESH

The Cartesian cut cell approach provides an alternative to traditional structured or unstructured body-fitted mesh. Using this approach meshes can be generated very efficiently using standard algorithms from computer graphics [17]. A uniform stationary background Cartesian mesh is employed, and the solid regions are simply cut out of the mesh. This results in the formation of three main types of cells, namely, the fluid cell, the solid cell and cut cell, see Figure 1. The use of cut cells leads to a boundary conforming representation of the geometry without the necessity of making the boundary a co-ordinate surface. Thus, no grid generation is performed in the conventional sense; all that is necessary is to calculate the intersections of a series of line segments with the background Cartesian mesh.

In order to generate the cut cells, the boundaries of solid regions are defined as polygons, each consists of a set of co-ordinate points which set up in an anti-clockwise direction. The number of points defining the body may be increased to generate smoother boundaries, but the underlying

DOI: 10.1002/fld

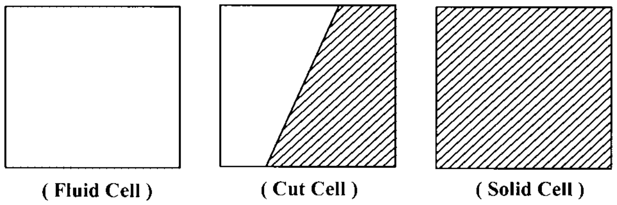


Figure 1. Three types of cells.

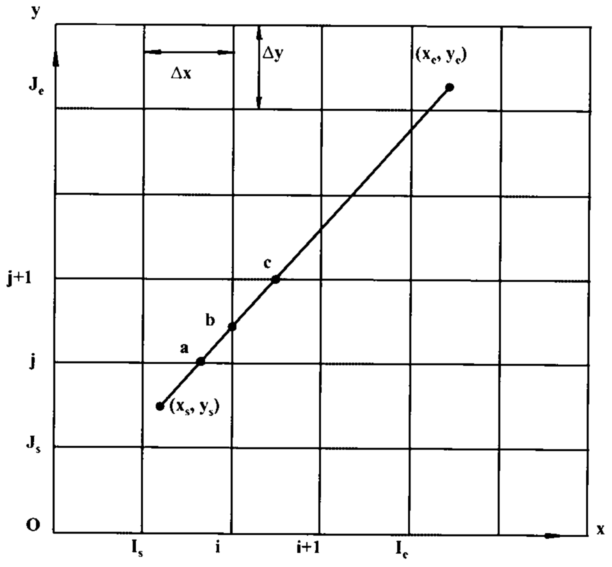


Figure 2. Finding intersection points of line segment.

definition is still considered piecewise linear. Furthermore, we impose the restriction that the minimum distance between two adjacent points, defining a single polygon, must be greater than the minimum mesh width. More than one solid region can be cut from the grid by specifying additional sets of points.

Suppose the intersection points of a particular straight line segment, defined in terms of its start and end co-ordinates (x_s, y_s) and (x_e, y_e) are to be found as follows. The address (I_s, J_s) of the cell containing the start point is given by

$$I_s = \text{int}\left(\frac{x_s - x_0}{\Delta x}\right) + 1 \quad \text{and} \quad J_s = \text{int}\left(\frac{y_s - y_0}{\Delta y}\right) + 1 \tag{4}$$

where x_0 and y_0 are the co-ordinates of the bottom left corner of the computational domain. The address (I_e, J_e) of the end point is found in a similar way. For convenience we identify which of the four quadrants $(0, 90°]$, $(90, 180°]$, $(180, 270°]$ or $(270, 360°]$ the slope of the line lies in.

Once the grid cells containing the start and end points of the line segment have been identified and its slope computed the required intersection points can be found. Figure 2 shows a line segment which cuts across a uniform background Cartesian mesh. The two intersection points of each cut cell must be located in the same background Cartesian cell. For the slope of the line is lied on $(0, 90°]$, the cut cell (i, j) has the entry intersection point **a** and the exit intersection point **b**. Meanwhile, the point **b** is the entry point of the next cut cell, and the exit intersection point is **c**. All the intersection points for subsequent grid cells intersected by the solid boundary line segment can be easily calculated. The cases where the slope of line lies in other quadrants can be treated analogously.

Once all intersection points have been found, the information necessary for the flow solver such as the normal direction of the outward pointing for each solid face, the centroid and side length normal, etc. for each cut cell can be easily obtained.

5. NUMERICAL SOLUTION

A requirement for treating the cut cells is that a finite volume approach is taken for discretizing the governing equations. Here, Equation (1) is integrated over an arbitrary control volume giving

$$\frac{\partial \mathbf{Q}_{i,j} V_{i,j}}{\partial \mathbf{t}} = -\oint_S \mathbf{F} \cdot \mathbf{n} \, \mathrm{d}s + \mathbf{B} V_{i,j} = -R(\mathbf{Q}_{i,j}) \tag{5}$$

where $\mathbf{Q}_{i,j}$ is average quantities at cell (i, j) stored at the cell centre, s and $V_{i,j}$ denote the boundary of the cell and area of cell (i, j), respectively. The surface integration on the right side of Equation (5) is evaluated by summing the flux vectors over each side of a cell, and the discrete form of the integral is

$$\oint_S \mathbf{F} \cdot \mathbf{n} \, \mathrm{d}s = \sum_{k=1}^{m} F_k \Delta l_k \tag{6}$$

where m is the number of the interfaces of cell (i, j); F_k is the numerical flux through interface k of cell (i, j); and Δl_k is the length of the interface. In the present study each fluid cell or solid cell has four interfaces, and each cut cell has three, four or five interfaces as shown in Figure 3.

5.1. Flux evaluation

In this study, Roe's flux splitting scheme is adopted to calculate the numerical inviscid fluxes. At each cell face, assuming a 1D Riemann problem in the direction normal to the cell face, the numerical inviscid fluxes can be expressed as follows:

$$F_k^I = \tfrac{1}{2}[F^I(Q_k^+) + F^I(Q_k^-) - |A|(Q_k^+ - Q_k^-)], \quad |A| = R|\Lambda|L \tag{7}$$

where $\mathbf{Q}_k^+$ and $\mathbf{Q}_k^-$ are the reconstructed right (or top) and left (or bottom) states at interface k of cell (i, j) and A is flux Jacobian matrix evaluated by Roe's average state. The quantities R and L

DOI: 10.1002/fld

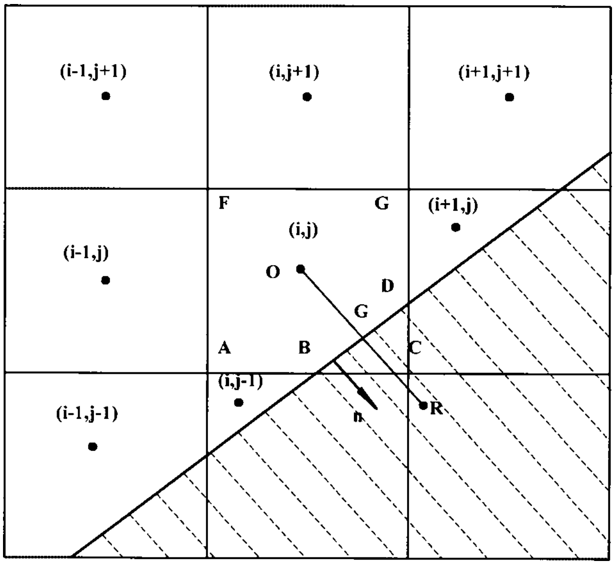


Figure 3. Calculation of gradients on the cut cell.

are right and left eigenvectors of A, Λ is the eigenvalues of A. The inviscid flux Jacobian A_{inv} is

$$A_{\text{inv}} = \frac{\partial(\mathbf{F}\cdot\mathbf{n})}{\partial\mathbf{Q}} = \begin{bmatrix} 0 & n_x & n_y & 0 \\ -u^2 n_x - uvn_y & 2un_x + vn_y & un_y & \beta n_x \\ -uvn_x - v^2 n_y & vn_x & un_x + 2vn_y & \beta n_y \\ -\dfrac{un_x}{\rho} - \dfrac{vn_y}{\rho} & \dfrac{n_x}{\rho} & \dfrac{n_y}{\rho} & 0 \end{bmatrix} \tag{8}$$

and its eigenvalues are given by

$$\lambda_{1,2} = un_x + vn_y, \quad \lambda_{3,4} = \tfrac{1}{2}(un_x + vn_y \pm c) \tag{9}$$

where $c = \sqrt{(un_x + vn_y)^2 + 4\beta/\rho}$.

To achieve second-order accuracy, a piecewise linear model for the cell variables must first be reconstructed from the stored cell centre data before the two Riemann states at each cell edge are computed. For a given cell with centre point (i, j), for example, this requires the reconstruction of the cell variables in the form

$$\mathbf{Q}(x, y) = \mathbf{Q}_{i,j} + \nabla\mathbf{Q}_{i,j} \cdot \mathbf{r} \tag{10}$$

where $\mathbf{r}$ is the vector from the cell centre to any point (x, y) within cell (i, j), $\mathbf{Q}_{i,j}$ is the cell centre data at the cell, and $\nabla\mathbf{Q}_{i,j}$ is the gradient of solution data at cell (i, j).

For flow cells which are away from a solid boundary, the gradient is evaluated using the known data at of neighbouring cell centres, and a slope limiter function is applied to prevent spurious

DOI: 10.1002/fld

overshoots or undershoots

$$\nabla \mathbf{Q}_{i,j} = G\left[\frac{\mathbf{Q}_{i+1,j}-\mathbf{Q}_{i,j}}{\Delta x}, \frac{\mathbf{Q}_{i,j}-\mathbf{Q}_{i-1,j}}{\Delta x}\right]\mathbf{n}_x + G\left[\frac{\mathbf{Q}_{i,j+1}-\mathbf{Q}_{i,j}}{\Delta y}, \frac{\mathbf{Q}_{i,j}-\mathbf{Q}_{i,j-1}}{\Delta y}\right]\mathbf{n}_y \tag{11}$$

where Δx and Δy are the uncut cell side lengths in the x and y directions, respectively. G is the slope limiter function.

In the present calculations, the limiter function is taken as

$$G(a,b) = s \cdot \max[0, \min(k|b|, s\cdot a), \min(|b|, ks\cdot a)] \tag{12}$$

where $s = \mathrm{sign}(b)$ and $1 \leqslant k \leqslant 2$. Differences in the choice of k are marginal in this scheme and do not affect its numerical stability. When $k=1$, Equation (12) reduces to the Minmod limiter; whilst, if $k=2$, Equation (12) is equivalent to Roe's Superbee limiter; in practice, $k=1.5$ has been found to give the best performance and was used in the present simulations.

For cut cells which have a solid boundary, the boundary condition should be taken into account for the calculation of the gradient [16]. By applying the solid wall boundary condition, the variables in fictional cell $\mathbf{R}$ can be obtained by (see Figure 3)

$$\begin{aligned} \rho_R &= \rho_{i,j} \\ \mathbf{v}_R &= -\mathbf{v}_{i,j} \\ p_R &= p_{i,j} - \rho_{i,j}\mathbf{g}\mathbf{n}_{\mathbf{y}}|RO| \end{aligned} \tag{13}$$

where $\mathbf{v} = u\mathbf{n}_{\mathbf{x}} + v\mathbf{n}_{\mathbf{y}}$. The gradient on cut cell (i,j) may be of two types: fluid and solid. We calculate the fluid gradients and solid gradients separately, i.e.

$$\begin{aligned} \mathbf{Q}_x^{\mathrm{f}} &= G\left(\frac{\mathbf{Q}_{i+1,j}-\mathbf{Q}_{i,j}}{\Delta x_{i+1/2,j}}, \frac{\mathbf{Q}_{i,j}-\mathbf{Q}_{i-1,j}}{\Delta x_{i-1/2,j}}\right) \\ \mathbf{Q}_y^{\mathrm{f}} &= G\left(\frac{\mathbf{Q}_{i,j+1}-\mathbf{Q}_{i,j}}{\Delta y_{i,j+1/2}}, \frac{\mathbf{Q}_{i,j-1}-\mathbf{Q}_{i,j}}{\Delta x_{i,j-1/2}}\right) \end{aligned} \tag{14}$$

where $\Delta x_{i+1/2,j} = x_{i+1,j} - x_{i,j}$, $\Delta y_{i,j+1/2} = y_{i,j+1} - y_{i,j}$ and

$$\begin{aligned} \mathbf{Q}_x^{\mathrm{s}} &= G\left(\frac{\mathbf{Q}_R-\mathbf{Q}_{i,j}}{\Delta x_{i,R}}, \frac{\mathbf{Q}_{i,j}-\mathbf{Q}_{i-1,j}}{\Delta x_{i-1/2,j}}\right) \\ \mathbf{Q}_y^{\mathrm{s}} &= G\left(\frac{\mathbf{Q}_{i+1,j}-\mathbf{Q}_{i,j}}{\Delta y_{i,j+1/2}}, \frac{\mathbf{Q}_{i,j}-\mathbf{Q}_R}{\Delta y_{j,R}}\right) \end{aligned} \tag{15}$$

where $\Delta x_{i,R} = x_R - x_{i,j}$, $\Delta y_{j,R} = y_{i,j} - y_R$ and G is the same slope limiter function as before.

Once the two types of gradients are calculated, a length average technique is used to obtained unique gradients in the cut cell,

$$\begin{aligned} \mathbf{Q}_x &= \frac{\Delta y_{\mathrm{s}}\mathbf{Q}_x^{\mathrm{s}} - \Delta y_{\mathrm{f}}\mathbf{Q}_x^{\mathrm{f}}}{\Delta y} \\ \mathbf{Q}_y &= \frac{\Delta x_{\mathrm{s}}\mathbf{Q}_y^{\mathrm{s}} - \Delta x_{\mathrm{f}}\mathbf{Q}_y^{\mathrm{f}}}{\Delta x} \end{aligned} \tag{16}$$

where $\Delta x_{\mathrm{f}} = |\mathbf{AB}|$, $\Delta x_{\mathrm{s}} = |\mathbf{BC}|$, $\Delta y_{\mathrm{f}} = |\mathbf{CD}|$ and $\Delta y_{\mathrm{s}} = |\mathbf{DE}|$.

Finally we can obtain the gradient vector

$$\nabla \mathbf{Q}_{i,j} = \mathbf{Q}_x \mathbf{n}_x + \mathbf{Q}_y \mathbf{n}_y \tag{17}$$

By involving the cut cell, the local accuracy will be between first- and second-order, but in practice it has been seen not to adversely affect the globe accuracy of this calculation method [16].

The viscous fluxes F_k^v are discretized using central difference approximations directly at a given cell face. The viscosity required at the cell face is computed by linearly interpolating between neighbour main cell centre values

$$\mu = \tfrac{1}{2}(\mu_i + \mu_{i+1}) \tag{18}$$

It should be noted that since the viscosity is assumed to be uniform within each fluid, the numerical value of μ at a given point can be derived from knowledge of the density distribution as follows.

First, define a parameter α as

$$\alpha \equiv \frac{\rho - \rho_2}{\rho_1 - \rho_2} \tag{19}$$

It can be seen that α is one in the first fluid and zero in the second fluid. Accordingly, the viscosity can be computed from

$$\mu = \alpha\mu_1 + (1-\alpha)\mu_2 \tag{20}$$

The viscous flux Jacobian A_{vis} is

$$A_{\text{vis}} = \frac{\partial(\mathbf{F}\cdot\mathbf{n})}{\partial \mathbf{Q}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \dfrac{\partial}{\partial x} n_x + \dfrac{\partial}{\partial \mathbf{y}} n_y & 0 & 0 \\ 0 & 0 & \dfrac{\partial}{\partial x} n_x + \dfrac{\partial}{\partial \mathbf{y}} n_y & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

The viscous fluxes over the solid edge of the cut cells can be directly evaluated similarly for the vertical and horizontal directions as

$$F_{\text{s}} \Delta L_{\text{s}} = \mu_{i,j} \mathbf{U}_{i,j} |BD|/|RO| \tag{22}$$

where $|BD|$ and $|RO|$ see, Figure 3.

5.2. *Integration in time*

By discretizing Equation (5) in time and omitting the subscripts for simplicity, the first-order Navier–Stokes implicit difference scheme can be expressed as

$$\frac{(\mathbf{Q}V)^{n+1} - (\mathbf{Q}V)^n}{\Delta t} = -R(\mathbf{Q}^{n+1}) \tag{23}$$

where V is the computational cell area. To achieve a time-accurate solution at each time step for unsteady flow problems, Equation (23) must be further modified in order to attain a divergence-free

DOI: 10.1002/fld

velocity field. This is accomplished by introducing a pseudo-time derivative into the system of equations, as

$$\frac{(\mathbf{Q}V)^{n+1,m+1}-(\mathbf{Q}V)^{n+1,m}}{\Delta\tau}+I_{ta}\,\frac{(\mathbf{Q}V)^{n+1,m+1}-(\mathbf{Q}V)^{n}}{\Delta t}=-R(\mathbf{Q}^{n+1,m+1}) \tag{24}$$

where τ is the pseudo-time and $I_{ta}=\mathrm{diag}[1,1,1,0]$. The right-hand side (RHS) of Equation (24) can be linearized using Newton's method at $m+1$ pseudo-time level to yield

$$\left[I_m V+\frac{\partial R(\mathbf{Q}^{n+1,m})}{\partial\mathbf{Q}}\right](\mathbf{Q}^{n+1,m+1}-\mathbf{Q}^{n+1,m})=-\left[I_{ta}\frac{(\mathbf{Q}^{n+1,m+1}-\mathbf{Q}^{n})V}{\Delta t}+R(\mathbf{Q}^{n+1,m})\right] \tag{25}$$

where

$$I_m=\mathrm{diag}\left[\frac{1}{\Delta\tau}+\frac{1}{\Delta t},\ \frac{1}{\Delta\tau}+\frac{1}{\Delta t},\ \frac{1}{\Delta\tau}+\frac{1}{\Delta t},\ \frac{1}{\Delta\tau}\right]$$

When $\Delta(\mathbf{Q}^{n+1,m})=\mathbf{Q}^{n+1,m+1}-\mathbf{Q}^{n+1,m}$ is iterated to zero, the density and momentum equations are satisfied, and the divergence of the velocity at time level $n+1$ is zero. In that case, the system of equations can be written in matrix form as

$$(D+L+U)\Delta\mathbf{Q}^{\mathrm{s}}=\mathrm{RHS} \tag{26}$$

where D is the block diagonal matrix; L is block lower triangular matrix and U is the block upper triangular matrix. Each of the elements in D, L and U is a 4×4 matrix. An approximate LU factorization (ALU) scheme as proposed by Pan and Lomax [23] can be adopted to form the inverse of Equation (25) in the form

$$(D+L)D^{-1}(D+U)\Delta\mathbf{Q}^{\mathrm{s}}=\mathrm{RHS} \tag{27}$$

Within each time step of the implicit integration the sub-iteration is terminated when the L_2 norm of the iteration process

$$L_2=\left\{\left[\sum_{i=1}^{N}(\mathbf{Q}^{\mathrm{s}+1}-\mathbf{Q}^{\mathrm{s}})^2\right]\Big/ N\right\}^{1/2} \tag{28}$$

is less than a specified limit ε. In the present study $\varepsilon=10^{-4}$.

6. NUMERICAL RESULTS

Two test cases have been calculated initially to show the feasibility and accuracy of the method for viscous flow problems. They are the unsteady Couette flow and the Rayleigh–Taylor instability problems. After that, the wave run up and over-topping on a smooth sea dike have been performed to show the applicability of the current method to complex free surface viscous flow problems. The time step Δt used for advancing the solution was within the range of 5×10^{-5}–5×10^{-4} s and the pseudo-time step was set as the order of 1000 times the real physical time step in order to accelerate convergence. All the calculations presented here are performed on a NEC SX-6i vector machine with four nodes. For a typical case using 50 × 150 grid points and 20 000 time steps, the computational time is around 2.5 h.

DOI: 10.1002/fld

6.1. Unsteady Couette flow

In order to evaluate the temporal accuracy of the new technique, we examine unsteady Couette flow. Two parallel plates are located a fixed distance, D, apart. The top plate moves at a constant velocity, $\mathbf{u} = (U_e, 0.0)$ and the bottom one is kept stationary. A periodic boundary condition is applied in the x-direction. The computational domain is shown in Figure 4.

In the present simulations, the fluid density ρ is 1.0 and the Reynolds number $Re = U_e D/v$ was taken to be 10. For this simple case, an analytical solution exists, namely

$$u(y,t) = U_e \frac{y}{D} + \sum_{m=1}^{\infty} \frac{2U_e(-1)^m}{\lambda_m D} \exp(-\upsilon\lambda_m^2 t)\sin\lambda_m y \qquad (29)$$

where υ is the kinematic viscosity of the fluid and $\lambda_{\mathbf{m}} = \mathbf{m}\pi/\mathbf{D}$, $m = 1, 2, 3, \ldots$.

A series of normalized velocity profiles at different times for this flow is shown in Figure 5. The numerical results are obtained using 20 cells in the y-direction. It can be seen that the numerical solutions are in excellent agreement with their analytic counterparts.

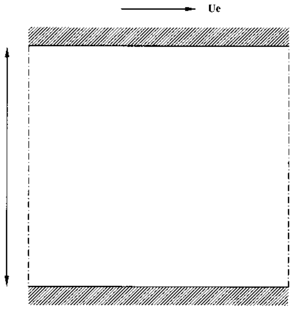


Figure 4. The computation domain for Couette flow.

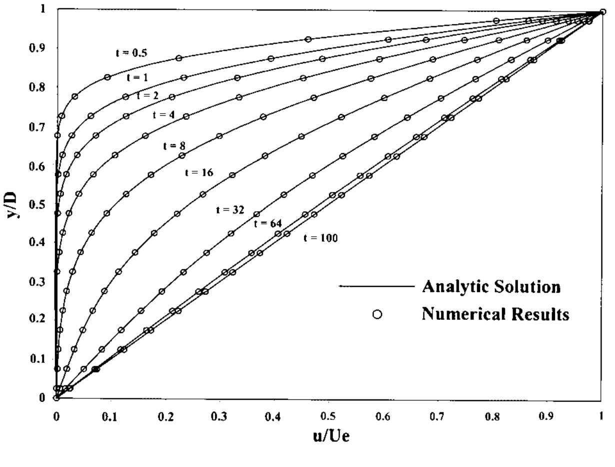


Figure 5. Normalized velocity profiles across the normalized channel width at different times.

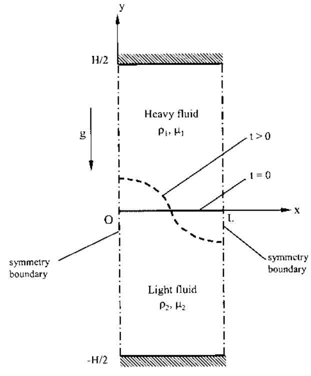


Figure 6. Illustration of the Rayleigh–Taylor instability problem.

6.2. Two-dimensional Rayleigh–Taylor instability

When a layer of heavy fluid is superposed over a layer of light fluid in a vertical gravitational field, the initial planar interface is unstable. Any disturbance will grow to produce spikes of heavy fluids moving downward and bubbles of light fluids moving upward. This phenomenon is known as the Rayleigh–Taylor instability.

The present work considers the two-dimensional Rayleigh–Taylor instability for a viscous incompressible case with two fluid layers. The kinematic viscosity is assumed to be the same for both heavy and light fluids, and the density ratio (ρ_1/ρ_2) for the fluids is set to be two. As illustrated in Figure 6, the fluids are confined within a rectangular domain of width $L = 0.02$ and height $H = 0.06$, which is bounded above and below by impermeable walls, and the interface of the two fluids coincides with $y = 0$. The flow field is assumed to be symmetric about the left and right boundaries. Following the work of Daly [24], a single wavelength perturbation is introduced at the fluid interface by using the following initial velocity field:

$$u = \begin{cases} \dfrac{\pi A \Delta y}{2L} \sin\left(\dfrac{\pi x}{L}\right) \exp\left(\dfrac{\pi |y|}{L}\right), & y \geqslant 0 \\ -\dfrac{\pi A \Delta y}{2L} \sin\left(\dfrac{\pi x}{L}\right) \exp\left(-\dfrac{\pi |y|}{L}\right), & y < 0 \end{cases} \tag{30}$$

$$v = \frac{\pi A \Delta y}{2L} \cos\left(\frac{\pi x}{L}\right) \exp\left(-\frac{\pi |y|}{L}\right)$$

where A is the perturbation amplitude and Δy is a representative mesh increment in the vertical direction. The velocity field corresponds to a sinusoidal perturbation of wavelength $2L$. To complete the specification of the initial conditions, the density field was prescribed as shown in Figure 6, while the initial pressure was set to a hydrostatic distribution initially.

On the case of unity perturbation amplitude A and Reynolds number of 28.3, three different uniform grid sizes $32 \times 96, 40 \times 120, 50 \times 150$ were chosen for a mesh refinement study [25]. The displacement of the interface at the left-wall was measured at $t = 6.0$ and the results for grid convergence are shown in Table I. Checking the solutions with Equation 5.10.5.2 [25], the result

Table I. Grid convergence results for Rayleigh–Taylor instability.

Grid	Displacement of the interface at the left wall	The order of convergence	Gird convergence index (%)
50×150	0.20497	1.8937	2.1965
40×120	0.20686	—	3.3209
32×96	0.20975	—	—

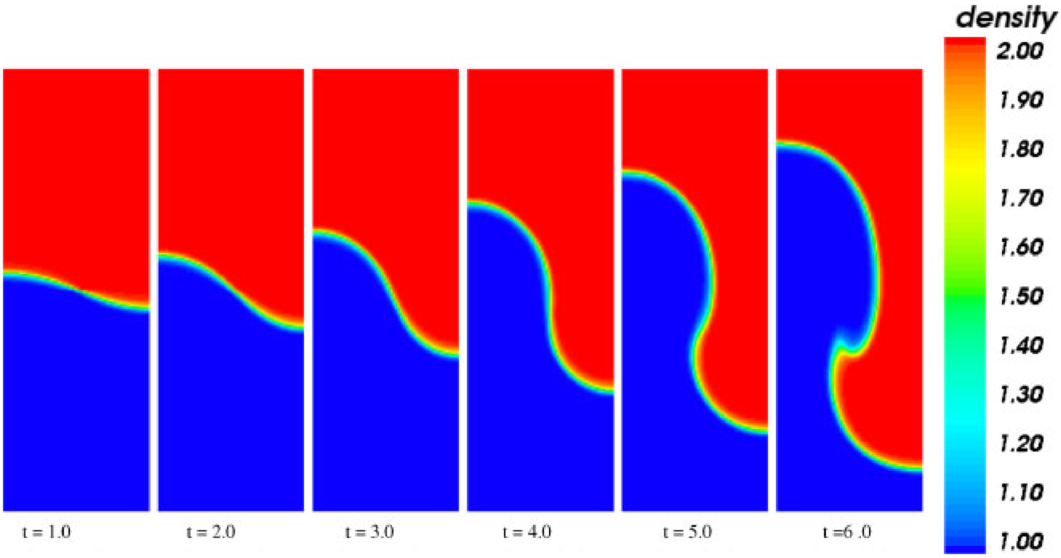


Figure 7. Free surface motion for Rayleigh–Taylor instability ($Re=28.3$).

is 1.0092, which means the solutions are well within the asymptotic range of convergence. For the following studies, the fine grid size 50×150 was used.

With unity perturbation amplitude A, two cases were run initially for Reynolds numbers 28.3 and 283. The computed evolution of the free surface is shown in Figures 7 and 8. In both the cases, the initial perturbation causes the light fluid to rise along the left boundary and the heavy fluid to sink along the right boundary. In the early stage of the computation, the movement of the interface is small and the displacement of the interface is seen to be symmetric. As time evolution continues, the amplitude of the instability increases and the characteristic mushroom shape emerges. The rollup of the interface is much more pronounced for the higher Reynolds number case due to the smaller influence of viscous effects, which would tend to smooth out sharp velocity gradients.

According to linear theory, the viscosity has a signification effect on the growth rate of the Rayleigh–Taylor instability, particularly in the case of short wavelength disturbances. For the specific case of two fluids with equal kinematic viscosities, Chandrasekhar [26] derived a relation between the linear growth rate and the perturbation wavelength. This variation can be expressed in terms of a non-dimensional growth rate $n^*=n\nu^{1/3}/\mathrm{g}^{2/3}$ and a modified Reynolds number $Re_\mathrm{m}=\lambda^{2/3}\mathrm{g}^{1/3}/\nu$, where n is the dimension growth rate (with unit of 1/s) and λ is the perturbation

DOI: 10.1002/fld

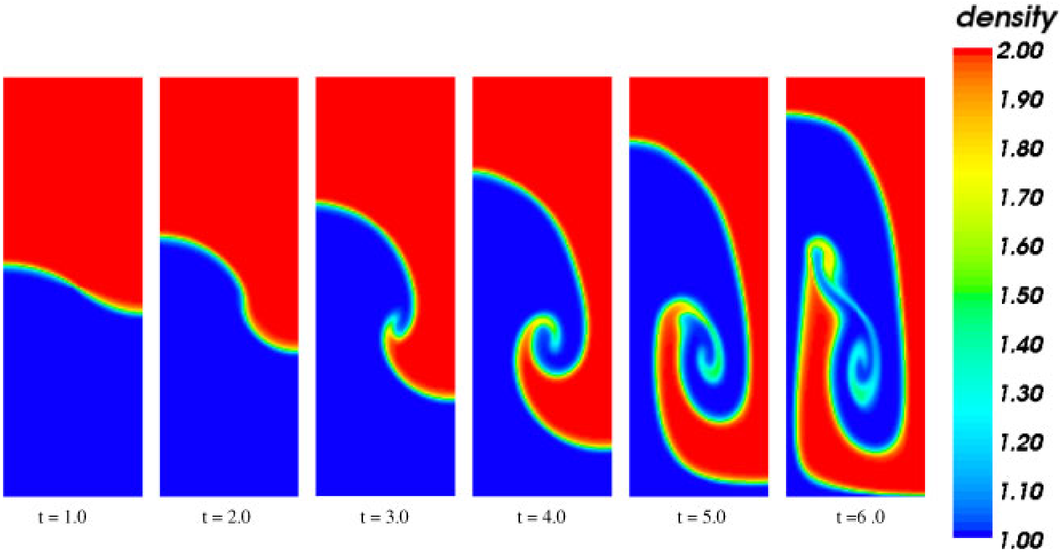


Figure 8. Free surface motion for Rayleigh–Taylor instability ($Re = 283$).

wavelength. From this result, the maximum growth rate and its corresponding wavelength can be determined for a given density ratio, kinematic viscosity and gravitational acceleration.

In order to determine whether the present method could reproduce the growth rates predicted by linear analysis, a series of calculations was performed for a density ratio of two using six different values of kinematic viscosity, corresponding to modified Reynolds numbers, Re_{m}, of 20, 40, 80, 200, 400 and 800. In these calculations the amplitude of the small perturbation, A, was taken to be 0.1 in order to maintain the linear behaviour of the solution over the duration of the simulation.

The growth rate was calculated from the numerical results using a procedure adapted by Daly [24]. A comparison of the computed non-dimensional growth rates with the variation predicted by the linear analysis is given in Figure 9. It can be seen that the numerical results are once again in good agreement with the theoretical results.

6.3. *Wave run up and over-topping of a smooth sea dike*

Troch *et al.* [27] simulated wave run-up and overtopping of a smooth impermeable sea dike. The 0.8 m tall dike with a 0.3 m wide crest stands in 0.7 m deep water, the seaward slope is 1:6 (Figure 10). It is important to note that the artificially low crest ensures that the majority of the waves overtop the structure. The computations were performed in a computational domain 6.3 m × 1.6 m which was discretized using a uniform 200 × 80 cells grid. Numerical wave gauges were positioned at 0.01, 1.00 m (co-located with the toe of the structure) and 3.81 m from the seaward boundary. The density ratio between water and air is taken as 1000:1. The kinematic viscosity of water and air are 1.003×10^{-6} and $1.8 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$, respectively, and the value of the gravitational acceleration is $9.81\,\mathrm{m/s}^2$. Figure 11 shows the Cartesian cut cell representation of the seaward slope of the dike.

DOI: 10.1002/fld

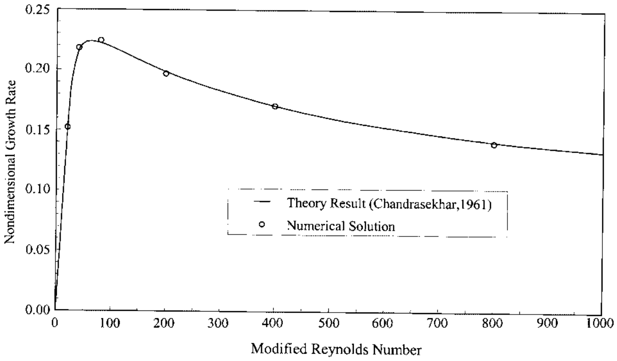


Figure 9. Non-dimensional perturbation growth rate *versus* modified Reynolds number for Rayleigh–Taylor instability.

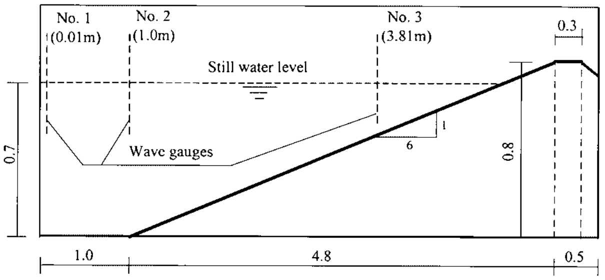


Figure 10. The computation domain for sea dike problem.

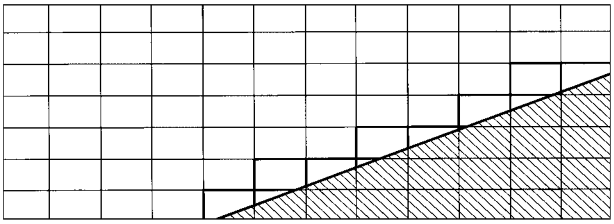
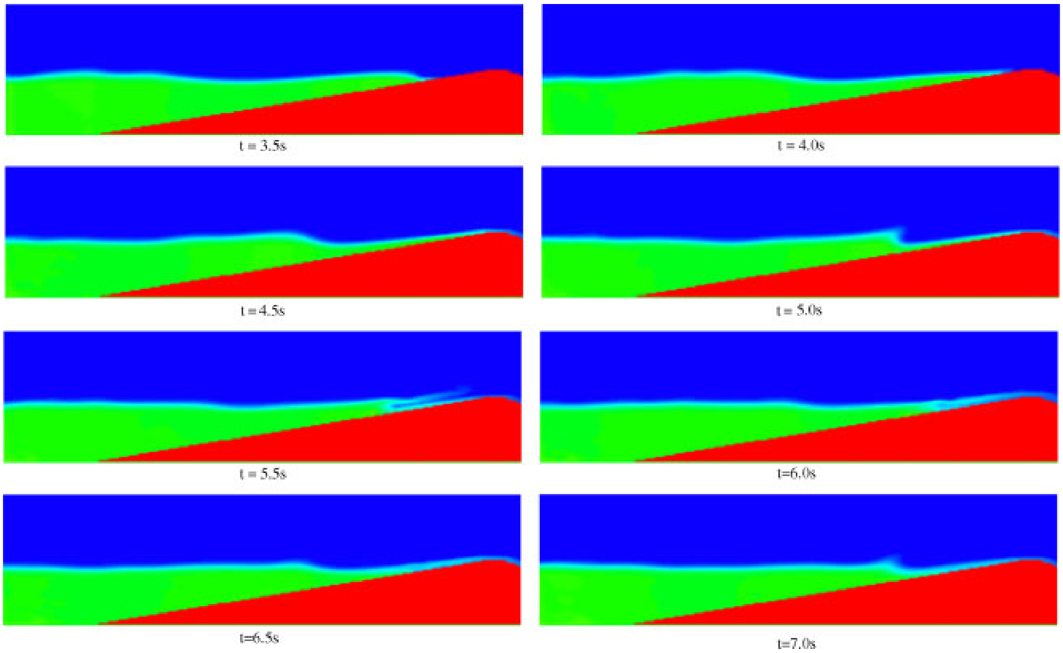


Figure 11. Local grid layout for the sloping sea dike (cut cells outlined in bold).

At the seaward boundary a series of random waves were introduced by specifying the local velocity and density distributions. These are calculated using linear wave theory, while the pressure is extrapolated from the interior of the flow domain. Using this theory, the instantaneous water velocity at a depth, z, below the free surface is found by linear superposition of N waves whose amplitude, wave number, and phase shift are determined by sampling an energy spectrum which


t = 3.5s

t = 4.0s

t = 4.5s

t = 5.0s

t = 5.5s

t=6.0s

t=6.5s

t=7.0s

Figure 12. Free surface motion for wave run-up and over-topping on a smooth sea-dike.

forms an appropriate model of the desired sea state, i.e.

$$\begin{aligned} u(x,z,t) &= \sum_{i=1}^{N} a_i\omega_i \frac{\cosh k_i(h+z)\sinh k_i h}{\sinh k_i h}\sin(\omega_i t - k_i x) \\ v(x,z,t) &= \sum_{i=1}^{N} a_i\omega_i \frac{\sinh k_i(h+z)\sinh k_i h}{\sinh k_i h}\cos(\omega_i t - k_i x) \end{aligned} \tag{31}$$

where a_i is the amplitude of the ith component, k_i is the wave number of the ith component, $\omega_i/(2\pi)$ is the frequency of the ith component and h is the local still water depth. ω and k are related by the, so-called, dispersion relationship $\omega^2 = gk\tanh(kh)$. In the present study, a modified JONSWAP spectrum [28] was chosen as the random wave spectrum at the seaward boundary:

$$\begin{aligned} S(f) &= \beta H_{1/3}^2 T_{\mathrm{p}}^{-4} f^{-5}\exp[-1.25(T_{\mathrm{p}}f)^{-4}]\cdot\gamma^{\exp[-(f/(f_{\mathrm{p}}-1))^2/2\sigma^2]} \\ \beta &= \frac{0.06238}{0.23+0.0336\gamma-0.185/(1.9+\gamma)}\cdot(1.094-0.01915\ln\gamma) \\ \sigma &= \begin{cases} 0.07, & f \leqslant f_{\mathrm{p}} \\ 0.09, & f > f_{\mathrm{p}} \end{cases} \end{aligned} \tag{32}$$

where f is the frequency, f_{p} and T_{p} denote the peak frequency and period, respectively, $H_{1/3}$ is the significant wave height (i.e. the 66th percentile of the wave-height distribution) and, $\gamma = 3.3$ is

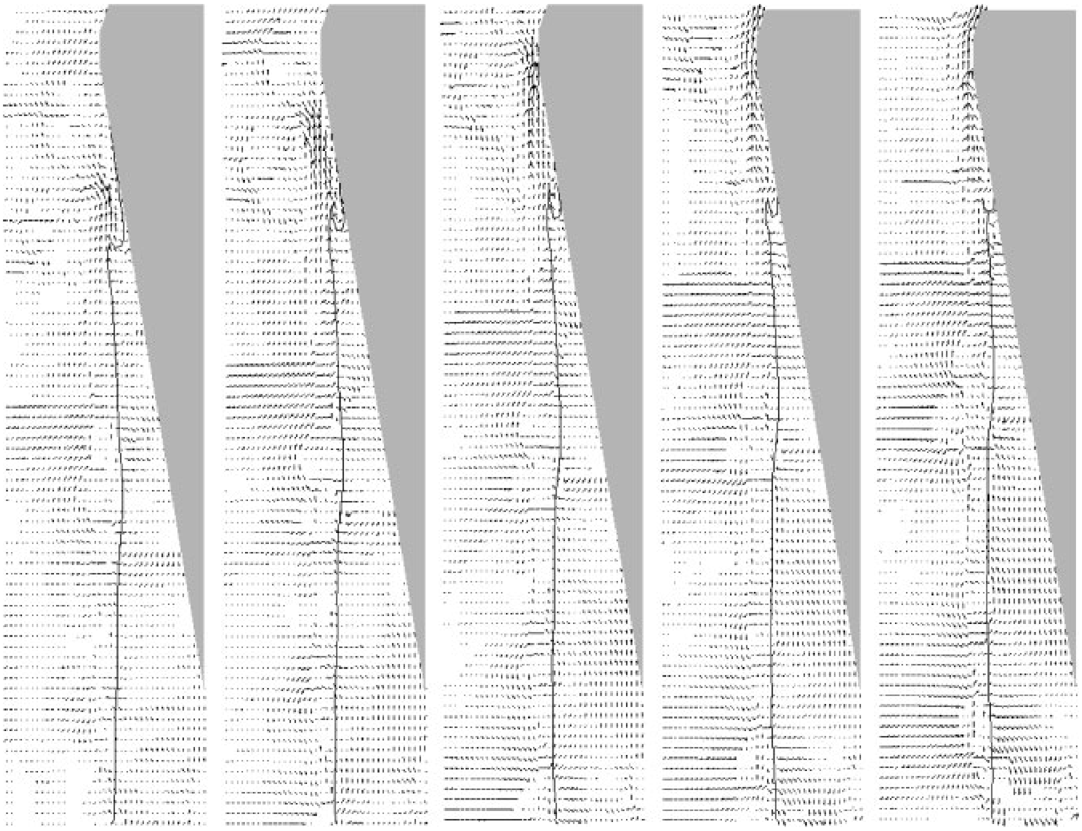


Figure 13. The velocity field for wave run-up and over-topping on a smooth sea-dike at $t = 5.2$, 5.4, 5.6, 5.8 and 6.0 s.

the peak enhancement factor. This spectrum may be used to generate waves characteristic of those occurring in shallower seas, and was developed as a result of long-term monitoring of the wave conditions in the North Sea [29].

The numerical simulations were conducted with $H_{1/3} = 0.16\,\text{m}$ and $T_{\text{p}} = 2\,\text{s}$ (defined at the seaward boundary), since the freeboard of the dike was only 0.10 m, these conditions mean that all significant waves will overtop the dike. Figure 12 shows the computed density profiles between 3.5 and 7.0 s, a period which corresponds to approximately two wave periods. As the waves start to run-up the dike shoaling occurs, causing the wave front to steepen and the wave height to increase, close to the location of the third gauge (3.81 m from the seaward boundary) the wave steepness may exceed the critical wave breaking criteria and the wave beings to break, forming a plunging breaker (Figure 13, $t = 5.5$ s). Lower amplitude waves, for example those shown in Figure 12 between 4.0 and 4.5 s, still overtop the dike but under these conditions green water overtopping occurs—where the water in the wave simply flows over the dike crest. These results show that the present method is able to simulate both kinds of overtopping events, whereas techniques which assume a single valued free surface (e.g. Boussinesq and shallow water models) which have commonly been used to simulate overtopping on this type of structure (e.g. Hedges and Reis [30]) cannot.

DOI: 10.1002/fld

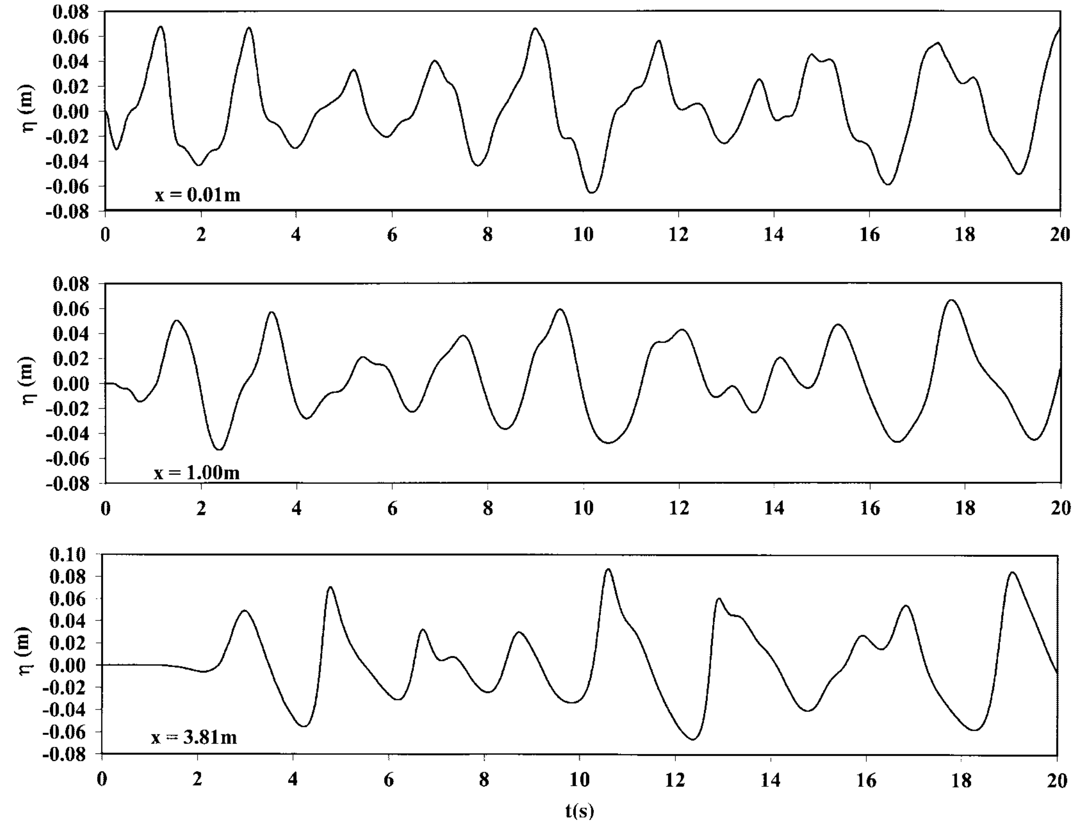


Figure 14. The time history of the free surface at three wave-gauge points.

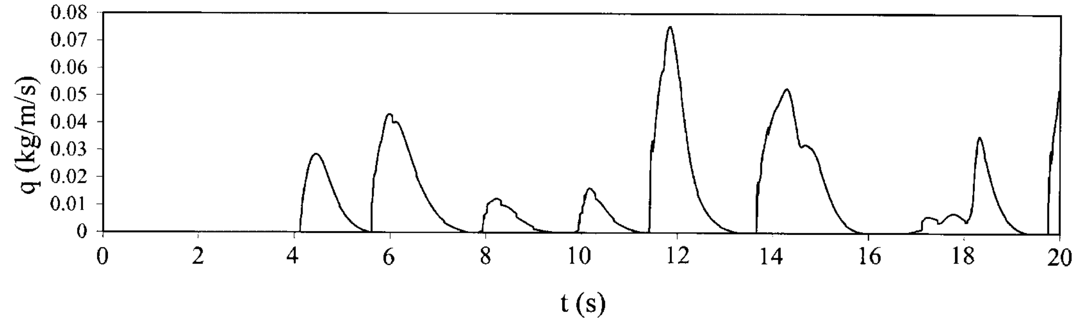


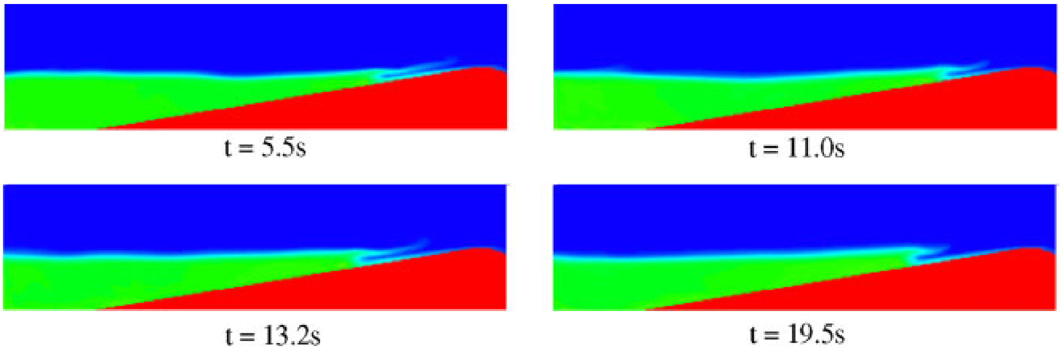
Figure 15. The time history of the instantaneous discharge at the middle of the dike crest.

Focusing on the plunging breaker at around $t = 5.5$ s, the velocity field from time 5.2 to 6.0 s are shown in Figure 13, which include both water and air. It can be seen clearly that the ultimate collapse of this wave causes a high velocity jet, which overtops the dike. The breaking processes

DOI: 10.1002/fld

Table II. Measured water mass discharging over the dike crest for each completed overtopping event during the simulation.

	Time (s)		
No.	*t*_start	*t*_end	Mass of water (kg/m)
1	4.13	5.57	4.71
2	5.60	7.67	9.52
3	7.84	9.75	2.12
4	9.86	11.32	2.21
5	11.42	13.32	12.78
6	13.66	15.91	13.83
7	16.90	19.35	5.32


t = 5.5s

t = 11.0s

t = 13.2s

t = 19.5s

Figure 16. The wave overturning profiles at four different times.

which lead up to this event cause the jet to entrain a significant amount of air, but the discharge over the dike associated with these events is still significant.

The time history of the free surface at the three wave gauges is shown in Figure 14. The character of the random wave and the effect of the slope at the wave are presented clearly. During the period between $t=0$ and 20 s, eight wave overtopping events occur with the last one still unfinished, which can be seen clearly in Figure 15. The completed overtopping and its last period are shown in Table II. Four plunging breakers are generated at around $t=5.5$, 11.0, 13.2, 19.5 s with wave breaking proceeding, which exhibit overturning free surface. The computed density profiles for these plunging breakers are shown in Figure 16.

7. CONCLUSIONS

A numerical scheme, based on the Cartesian cut cell approach, for the two-dimensional Navier–Stokes equations for unsteady, incompressible, fluid flows with variable density has been developed and tested. This method captures the interface between two fluids automatically as part of the solution without special provision along with the other flow variables. Complex geometries arising in real flow problems are represented by using the Cartesian cut cell method. An implicit dual-time iteration technique allows the equations to be solved as a hyperbolic system in a pseudo-time and accurate solutions obtained at each real physical time step. In this paper, both unsteady Couette flow and Rayleigh–Taylor instability have been successfully simulated, giving confidence in the capabilities of modelling the low Reynolds number calculations. The solver has then been applied to compute wave run-up and subsequent overtopping at a smooth impermeable sea dike. These results show that the solver may be applied to highly transient free surface calculations of the type often found in coastal engineering.

ACKNOWLEDGEMENTS

This work was financially supported by the (EPSRC) (U.K.) under the grant (GR/T18622/01). Dr Ingram wishes to acknowledge the support of the Scottish Funding Council for his position and for their support of the Joint Research Institute with the Heriot-Watt University as a component of the Edinburgh Research Partnership.

REFERENCES

1. Farm J, Martinelli L, Jameson A. Fast multigrid method for solving incompressible hydrodynamic problems with free surfaces. *AIAA Journal* 1994; **32**:1175–1182.
2. Thé JL, Raithby GD, Stubley GD. Surface-adaptive finite-volume method for solving free-surface flows. *Numerical Heat Transfer*, *B*: *Fundamentals* 1994; **26**:367–380.
3. Mayer S, Garapon A, Sørensen L. Finite volume solutions to unsteady free surface flow with application to gravity waves. *Coastal Dynamics '97*. Plymouth, 1997; 118–127.
4. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1967; **8**:2182–2189.
5. Hirt CW, Nichols BD. Volume of fluid (VoF) methods for dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
6. Youngs DL. Time dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: London, 1982; 237–285.
7. Li T, Troch P, De Rouck J. A solver for numerical simulation of breaking waves using a cut-cell VOF cell-staggered finite-volume approach. *Technical Report*, Department of Civil Engineering, Ghent University, Belgium, 2003.
8. Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics* 1999; **153**:26–50.
9. Unverdi SO, Tryggvason G. A front tracking method for viscous, incompressible multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.
10. Sussman M, Smereka P, Osher S. A level set approach for computing solution to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
11. Kelecy FJ, Pletcher RH. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *Journal of Computational Physics* 1997; **138**:939–980.
12. Pan D, Chang CH. The capturing of free surfaces in incompressible multi-fluid flows. *International Journal for Numerical Methods in Fluids* 2000; **33**:203–222.
13. Qian L, Causon DM, Ingram DM, Mingham CG. A Cartesian cut cell two-fluid solver for hydraulic flow problems. *Journal of hydraulic Engineering* (ASCE) 2003; **129**(9):688–696.
14. Qian L, Causon DM, Ingram DM, Mingham CG. A free-surface capturing method for two fluid flows with moving bodies. *Proceedings of the Royal Society*, *Series A* 2006; **462**:21–42.
15. Yang G, Causon DM, Ingram DM, Saunders R, Batten P. A cartesian cut sell method for compressible flows. Part A. Static Body Problems. *Aeronautical Journal* 1997; **101**(1101):47–56.
16. Causon DM, Ingram DM, Mingham CG, Pearson RV. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advances in Water Resources* 2001; **24**:899–911.

DOI: 10.1002/fld

17. Ingram DM, Causon DM, Mingham CG. Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation* 2003; **61**:561–572.
18. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
19. Rogers SE, Kwak D. Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. *AIAA Journal* 1990; **28**:253–262.
20. Rogers SE, Kwak D, Kiris C. Steady and unsteady solutions of the incompressible Navier–Stokes equations. *AIAA Journal* 1991; **29**:603–610.
21. Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*, *AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, HI, June 1991.
22. Anderson WK, Rausch RD, Bonhaus DL. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *Journal of Computational Physics* 1996; **128**:391–408.
23. Pan D, Lomax H. A new approximate LU factorisation scheme for the Navier–Stokes equations. *AIAA Journal* 1988; **26**:163–171.
24. Daly BJ. Numerical study of two fluid Rayleigh–Taylor instability. *Journal of Computational Physics* 1967; **10**:297–307.
25. Roache PJ. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers: Albuquerque, 1998.
26. Chandrasekhar S. *Hydrodynamics and Hydromagnetics Stability*. Oxford University Press: London, 1961; 428–447.
27. Troch P, Li T, De Rouke J, Ingram DM. Wave interaction with a sea dike using a VOF finite volume method. *Proceedings of the 13th International Offshore and Polar Engineering Conference*, vol. 3, 2003; 325–332.
28. Goda Y. A comparative review on the functional forms of directional wave spectrum. *Coastal Engineering Journal* 1999; **41**(1):1–20.
29. Hasselmann K, Barnett TP, Bouws E, Carlson H, Cartwright DE, Enke K, Ewing JA, Gienapp H, Hasselmann DE, Kruseman P, Meerburg A, Mller P, Olbers DJ, Richter K, Sell W, Walden H. Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). *Ergnzungsheft zur Deutschen Hydrographischen Zeitschrift Reihe A* 1973; **8**(12):95.
30. Hedges TS, Reis MT. Accounting for random wave run-up in overtopping predictions. *Maritime Engineering*, *Proceedings of the ICE* 2004; **157**(3):113–122.

DOI: 10.1002/fld