INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 931–952 (1997)

A MIXED LAGRANGIAN–EULERIAN APPROACH TO MODELLING FLUID FLOW DURING MOULD FILLING

R. W. LEWIS,* S. E. NAVTI AND C. TAYLOR

Institute for Numerical Methods in Engineering, University of Wales, Swansea SA2 8PP, U.K.

SUMMARY

An updated free surface Lagrangian–Eulerian finite element kinematic description is used to simulate free surface flow problems associated with mould filling. The method proposed results in an accurate determination of the front, making it ideal for problems in which free surface boundary conditions play an important role. Significant saving in CPU time can be obtained over other fixed mesh approaches by virtue of the air domain being ignored. Assuming a laminar regime for the flow field, a mixed interpolation formulation is used to approximate the discretized governing equations for elimination.

Of particular interest is the method implementation to restrict the number of remeshing operations and track the moving free surface within an arbitrary domain, either with or without internal obstacles. The method used to automatically assign boundary conditions to the changing domain is described. A dam break problem is modelled numerically and compared against experimentally derived data in order to validate the model. A further numerical example demonstrates the capabilities of the algorithm developed to model the filling of an industrial casting. © 1997 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 931–952 (1997)

No. of Figures: 23. No. of Tables: 0. No. of References: 38.

KEY WORDS: finite element method; free surface; updated Lagrangian–Eulerian

1. INTRODUCTION

Numerical modelling of free surface flow has many important industrial applications. These include applications in environmental engineering, mould filling in metal casting, injection moulding through to forging, rolling, extrusion and many other processes. The prediction of flow patterns for problems in which the free surface boundary condition are important requires an accurate definition of that surface for solutions to be acceptable. There currently exist a number of methods to model free surface flow problems. These can be conveniently divided into two categories, the fixed and moving mesh types.

The natural choice for moving boundary problems would be of the Lagrangian and arbitrary Lagrangian–Eulerian types.[1–3] Both methods allow for an explicit determination of the free surface, with the front ideally located on that boundary.

In the updated Lagrangian approach the mesh for a given time step is incremented at the free surface in terms of the instantaneous surface velocity field. The main drawback of this method is that the mesh deforms severely as the free surface moves, making remeshing a must at almost all time

* Correspondence to: R. W. Lewis, Institute for Numerical Methods in Engineering, University of Wales, Swansea, Singleton Park, Swansea SA2 8PP, U.K.

CCC 0271–2091/97/080931–22 $17.50 *Received November 1996*

steps. A robust remeshing algorithm is therefore essential for the success of this method. Special treatments are also required to transfer the geometry definition of the fluid domain into the finite element mesh for the next time step. An attractive aspect of this approach, however, is that it offers certain advantages when dealing with advection-dominated transport equations. A similar front-tracking model has been developed by Muttin *et al.*[4]

The arbitrary Lagrangian–Eulerian formulation is a variant of the Lagrangian formulation, with the mesh deforming in terms of an arbitrary velocity, independent of the flow velocities, expect at the free surface. The speed of the moving mesh is incorporated into the convective term, making it more stable numerically and significantly reducing mesh distortions. The selection of the mesh velocity is, however, not trivial and requires an experienced user for proper implementation. Other authors have tracked the free surface along preselected splines. However, owing to the complex nature of the filling patterns in mould filling, the above method would be difficult to implement, since the generation of appropriate splines requires some knowledge of the filling pattern.[5,6] The moving mesh approached has additional requirements regarding the treatment of contact with the enclosing wall and it also introduces truncation errors during the interpolation of variables from the distorted mesh to the new mesh.

In the second category a fixed mesh is generated to cover the spatial domain and the free surface is augmented by making use of a nodal variable or function, which is advected using velocities obtained from the Navier–Stokes equations. The nodal variable or fill factor (pseudoconcentration factor) determines the position of the free surface by giving an indication of the percentage of the mould filled. Two fluids are modelled, namely flowing fluid of interest and a fictitious material in the empty region. This method requires no geometry manipulations for the free surface after the finite element mesh is generated and it can easily be applied to the treatment of complex geometries. However, it results in a smeared free surface and, owing to the incompressibility condition imposed, entrapped air cannot escape. Temporary holes and a Darcy-type law have been used to allow air to filter through the porous walls. In both cases an experienced user is needed for the resulting filling pattern to be realistic. Another type in this category is named fringe element generation. In this method, fringe elements are generated temporarily at the free surface so boundary conditions are more accurately satisfied. Some authors have used markers distributed in the fluid domain as material points (MAC).[7,8] A net inflow method has been applied to simulate time-dependent free surface flows.[9] In this method both filled and unfilled domains are included in the analysis, with the volume of the incompressible liquid in each control volume calculated at each iteration. Finite element formulations have also been developed for simulating metal-casting flows using the fixed mesh approach.[10–23]

The method advocated for modelling free surface flow in geometries where free surface boundary conditions are important is the mixed Eulerian–updated Lagrangian free surface method. The model is based on an Eulerian finite element formulation combined with a Lagrangian free surface incrementation method. The principal advantage of this method lies in the natural representation of the free surface. Owing to the explicit nature of the free surface incrementation process, an accurate description of the front results. As the mesh deforms, various methods are used to prolong its useful life. However, when the mesh becomes too distorted, remeshing has to be initiated. The efficiency of the mesh generator is crucial in ensuring that the time saved in omitting computations in the air domain is not used up in remeshing the domain. In two dimensions the CPU time required for mesh generation is small compared with the solution time.

Other authors have applied finite difference schemes together with the Sola-VOF method.[24,25]

2. PHYSICAL MODEL

In the following sections the governing equations and boundary conditions are presented.

2.1. Hydrodynamics

The general problem for an incompressible fluid is taken into account here. The governing equations are the incompressibility condition

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

defined on Γ_{me} and the transport of fluid momentum

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}\right) - \tau + \nabla p = \mathbf{f} \tag{2}$$

defined on Γ_{me}. Here, $\mathbf{u}$ is the fluid velocity, ρ is the density, p is the scalar pressure, $\mathbf{f}$ is the body force (ρg) and τ is the viscous stress tensor given by

$$\tau = 2\mu S, \tag{3}$$

with

$$S = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\text{T}}], \tag{4}$$

where μ is the dynamic viscosity and S is the rate-of-strain tensor.

3. BOUNDARY CONDITIONS

As indicated in Figure 1, two main subsets of free surface boundary conditions were employed on Γ_{f} (fluid–air interface) and Γ_{s} (fluid–solid boundary interface) respectively. These take the form of specified velocities $\mathbf{u}$ and pressure p on the boundaries,

$$\mathbf{u} = f(x, y) \quad \text{defined on } \Gamma_{\text{s}}, \qquad p = f(x, y) \quad \text{defined on } \Gamma_{\text{f}}, \tag{5}$$

where Γ_{s} is the solid wall boundary and Γ_{f} is the free surface boundary.

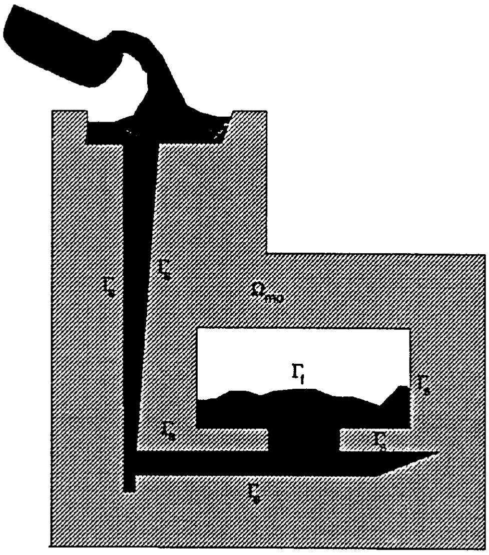


Figure 1. Problem definition

The free surface was treated as traction-free with pressure specified as atmospheric. Slip or no-slip boundary conditions could be implemented at the solid wall boundaries. For no-slip boundary conditions all the nodes on the solid wall are fixed, i.e. $u_n = 0$ and $u_\tau = 0$, except for the nodes that lie on the free surface, for which a slip boundary condition was applied, so only $u_n = 0$. In specifying slip boundary conditions on the solid boundaries, the effects of the boundary layer on the flow are effectively assumed to be negligible during the small step used. This is the case for all the numerical examples discussed in this paper.

Use has been made of a method implemented by Engelman *et al.*[26] and Pinder and Gray[27] to cope with flow in domains where some of the flow boundaries do not correspond to the global Cartesian axes. Equations at the relevant boundary nodes, which are not on the free surface or at the inlet, are transformed in the local directions and tangential and/or normal boundary conditions are applied as desired. By premultiplying the unknowns in the global direction by a rotation matrix, the unknowns in the local direction can be obtained. The force vector can be determined in a similar manner. This can be written as

$$\mathbf{U}_\mathrm{L} = \mathbf{R}\mathbf{U}_\mathrm{G}, \qquad \mathbf{F}_\mathrm{L} = \mathbf{R}\mathbf{F}_\mathrm{G}, \tag{6}$$

where $\mathbf{U}_\mathrm{L}$ and $\mathbf{U}_\mathrm{G}$ are vectors holding unknowns in the local and global directions, $\mathbf{F}_\mathrm{L}$ and $\mathbf{F}_\mathrm{G}$ are the force vectors in the local and global directions and $\mathbf{R}$ is the rotation matrix. In two dimensions this is made up to 2×2 submatrices on its diagonal. For each relevant boundary node this matrix can be written as

$$\begin{pmatrix} n_y & -n_x \\ n_x & n_y \end{pmatrix},$$

where n_x and n_y are direction cosines with respect to the global directions x and y. The rotation matrix for nodes where no rotation of the axes is required is simply a 2×2 identity matrix.

For nodes where transformation to the local co-ordinate direction is desired, the complete transformation matrix changes from $\mathbf{K}_\mathrm{G}\mathbf{U}_\mathrm{G} = \mathbf{F}_\mathrm{G}$ in the global direction to $\mathbf{R}\mathbf{K}_\mathrm{G}\mathbf{R}^\mathrm{T}\mathbf{U}_\mathrm{L} = \mathbf{R}\mathbf{F}_\mathrm{G}$ in the local direction. This is obtained bearing in mind the orthogonal nature of the rotational matrix $\mathbf{R}$.

The direction cosines for the boundary segments are evaluated before the solution process using the expressions

$$n_{x_{ij}} = \frac{y_j - y_i}{\sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}}, \qquad n_{y_{ij}} = \frac{x_j - x_i}{\sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]}},$$

where (x_i, y_i) and (x_j, y_j) are the co-ordinates of the end points of the boundary segment ij, Figure 2(i). The normal for the boundary node at (x_k, y_k), Figure 2(ii), is evaluated in a similar manner.

Nodes that lie on any boundary segment or boundary node are assigned the normals that correspond to it. Other nodes, typically lying on the front, have their direction cosines evaluated based on the 'consistent' normals, i.e. mass conservation arguments, and can be written as

$$n_{x_i} = \frac{1}{n_i}\int_\Omega \frac{\partial N_j}{\partial x}\mathrm{d}\Omega, \qquad n_{y_i} = \frac{1}{n_i}\int_\Omega \frac{\partial N_j}{\partial y}\mathrm{d}\Omega,$$

with

$$n_i = \sqrt{\left[\left(\int_\Omega \frac{\partial N_j}{\partial x}\mathrm{d}\Omega\right)^2 + \left(\int_\Omega \frac{\partial N_j}{\partial y}\mathrm{d}\Omega\right)^2\right]}, \tag{7}$$

where only elements that contain the node i contribute to the integrals. N represents quadratic shape functions for the elements in question over the domain Ω.

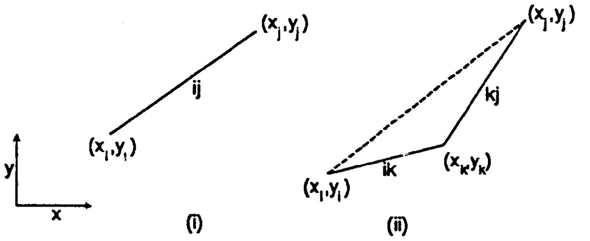


Figure 2. Co-ordinate used for evaluation of normals

The rotation matrices are then assembled from the evaluated normals, with the signs of the direction cosines reversed to direct them into the element.

4. NUMERICAL MODEL

4.1. Finite element formulation

In order to obtain the discretized form of the equations, the governing equations are spatially discretized[28–30] using the conventional Galerkin weighted residual technique.[31]

A mixed interpolation formulation was adopted[32] enabling us to approximate the discretized differential equations so that the variables can be reduced by elimination.

Six-noded triangular elements which satisfy the Brezzi–Babuska condition, Figure 3, i.e. ● u, v, p and ○ u, v,[29] were used. These have the advantage of optimal convergence rate and have been shown to have a proven record of performance in problems of incompressible flow.[33] The above element was also chosen owing to the flexibility it offers for Lagrangian moving mesh problems.

4.2. Spatial discretization

Shape functions are used for approximating the field variables over the domain. In a typical element e the velocities and pressure are given by

$$u_e = \sum_{i=1}^{n} N_i u_i, \qquad v_e = \sum_{i=1}^{n} N_i v_i, \qquad p_e = \sum_{i=1}^{n'} N'_i p_i, \tag{8}$$

where n is the number of velocity nodes in the element e and n' is the number of pressure nodes. Similarly, N represents quadratic shape functions for velocity interpolation and N' represents linear shape functions for pressure interpolation.

An expression for the residual can be obtained using equations (1) and (2). Minimization is then required using weighting functions. Applying Green's theorem to the momentum equation derived, the weak form of the weighted residual equation is obtained.

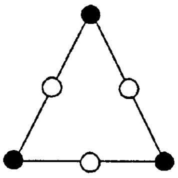


Figure 3. Element used

In the continuity equation the pressure shape functions N_i' are used for the weighting functions, as the continuity equation will only be enforced at the pressure nodes. The final set of spatially discretized equations may be written in the form

$$\mathbf{M}\dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta} = \mathbf{F}, \tag{9}$$

where **M** is analogous to the mass matrix, **K** is the convective and diffusive matrix, **F** is the force vector and θ is the field vector.

In matrix form,

$$\begin{pmatrix} \rho\mathbf{M}_u & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \rho\mathbf{M}_v \end{pmatrix}\begin{pmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \end{pmatrix} + \begin{pmatrix} \mathbf{K}_{uu} & \mathbf{C}_u & \mathbf{K}_{uv} \\ \mathbf{C}_u^{\mathrm{T}} & \mathbf{0} & \mathbf{C}_v^{\mathrm{T}} \\ \mathbf{K}_{vu} & \mathbf{C}_v & \mathbf{K}_{vv} \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_u \\ 0 \\ \mathbf{F}_v \end{pmatrix}, \tag{10}$$

where

$$\mathbf{M}_u = \mathbf{M}_r = \int_\Omega N_i N_j,$$

$$\mathbf{K}_{uu} = \mathbf{K}_{vv} = \int_\Omega \rho\left(N_i N_k u_k \frac{\partial N_j}{\partial x} + N_i N_k v_k \frac{\partial N_j}{\partial y}\right) + \mu\left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right),$$

$$\mathbf{K}_{uv} = \mathbf{K}_{vu} = 0$$

are matrices of the order $n \times n$ (where n is the number of velocity interpolation nodes),

$$\mathbf{C}_u = -\int_\Omega \frac{\partial N_i}{\partial x} N_j', \qquad \mathbf{C}_v = -\int_\Omega \frac{\partial N_i}{\partial y} N_j' \tag{11}$$

$n \times n'$ matrices,

$$\mathbf{C}_u^{\mathrm{T}} = -\int_\Omega N_i' \frac{\partial N_j}{\partial x}, \qquad \mathbf{C}_v^{\mathrm{T}} = -\int_\Omega N_i' \frac{\partial N_j}{\partial y} \tag{12}$$

are $n' \times n$ matrices, the n-dimensional force vectors are

$$\mathbf{F}_u = -\int_\Gamma N_i f_x, \qquad \mathbf{F}_v = -\int_\Gamma N_i f_y \tag{13}$$

and ρ is the density.

4.3. Temporal discretization

Finite difference approximations are used to spatially discretize the resulting first-order ordinary differential equation with respect to time.

From the generalized midpoint family of methods[34,35] the differential equation $\mathbf{M}\dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta} = \mathbf{F}$ can be written as

$$\mathbf{M}(\theta_{n+\alpha}, t_{n+\alpha})\dot{\boldsymbol{\theta}}_{n+\alpha} + \mathbf{K}(\boldsymbol{\theta}_{n+\alpha}, t_{n+\alpha})\boldsymbol{\theta}_{n+\alpha} = \mathbf{F}_{n+\alpha}, \tag{14}$$

where

$$\boldsymbol{\theta}_{n+1} = (1-\alpha)\boldsymbol{\theta}_n + \alpha\boldsymbol{\theta}_{n+1}, \qquad \dot{\boldsymbol{\theta}}_{n+1} = \left(\frac{\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n}{\Delta t}\right), \qquad t_{n+1} = t_n + \alpha\Delta t. \tag{15}$$

Substituting (18) in (17), we obtain the expression

$$\left(\frac{\mathbf{M}_{n+\alpha}}{\Delta t} + \alpha\mathbf{K}_{n+\alpha}\right)(\boldsymbol{\theta}_{n+\alpha}) = \left(\frac{\mathbf{M}_{n+\alpha}}{\Delta t} - (1-\alpha)\mathbf{K}_{n+\alpha}\right)(\boldsymbol{\theta}_n) + (\mathbf{F}_{n+\alpha}). \tag{16}$$

For different values of α, different members of this family of methods can be identified. For the examples presented later, the backward difference method is used. This results in the expression

$$[\mathbf{M}_{n+1} + \Delta t\mathbf{K}_{n+1}](\boldsymbol{\theta}_{n+1}) = [\mathbf{M}_{n+1}](\boldsymbol{\theta}_n) + (\Delta t\mathbf{F}_{n+1}). \tag{17}$$

4.4. *Iterative procedure*

The set of non-linear equations specified in (9) is solved using a simple relaxation procedure.[31]

1. Initial values are assumed for the primitive variables.
2. These are solved for the updated values u_{n+1}, v_{n+1} and p_{n+1}.
3. $(u_{n+1} - u_n)/u_{n+1}$, $(v_{n+1} - \tilde{v}_n)/v_{n+1}$ and $(p_{n+1} - \tilde{p}_n)/p_{n+1}$ at all nodal points are evaluated. If within a specified tolerance, the solution has converged.
4. If the difference in any of the above expressions is greater than or equal to the tolerance, $\tilde{u}$ is updated to a simple arithmetic mean $(u_{n+1} + u_n)/2$ with a weighting factor of 0·5. The whole process is repeated until convergence is achieved.

4.5. *Equation solution technique*

The method adopted for solving the assembled matrix equation is based on the direct elimination frontal solution method.[31] This is based on direct Gaussian elimination for solving symmetric matrices, with the leading diagonal always used as a pivot.

Upon remeshing the domain, a front width minimization module is used to renumber the elements, thereby minimizing the maximum number of equations stored in the core at any one time step.

5. GEOMETRY DEFINITION

5.1. *Overview*

Prior to initiating the finite element analysis, the geometrical model has to be defined. This would typically contain information on the metal–mould and metal–air interfaces. Boundary specification is currently being done during the preprocessing part of the mesh generation process. The defined boundary consists of a collection of straight and/or curved wall segments forming one or multiple enclosed loops. This is defined in a boundary control file.

5.2. *Discretization*

In order to accurately define the geometry of a complex casting with curved boundaries, a large number of interconnected straight boundary segments would be required, Figure 4. To improve the

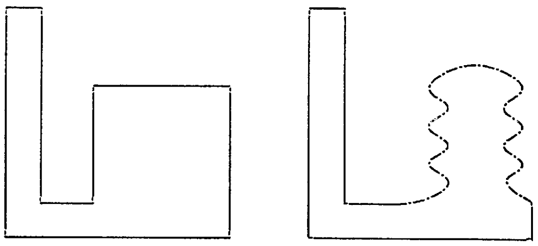


Figure 4. Boundary segments

geometry definition process while minimizing the number of boundary nodes or segments required, three boundary segment types were chosen.

The first type consisted of simple straight line segments with specified end points. This is suitable for most simple castings with straight line boundaries. The second type was obtained by defining the centre point (x_0, y_0) of a circular arc, its radius (r), start point (x_1, y_1) and end point (x_2, y_2). The third case is obtained by specifying key points and fitting a cubic spline through them. This typically results in significant improvement in the geometrical model for highly curved boundaries. Figure 5 shows a schematic representation of the different types of segments used. Once the boundary segments are defined, they are discretized by subdividing them to form a closed loop of interconnecting line segments. The nomenclature adopted is anticlockwise, to enable us determine, during the front update process, in which domain the node lies.

5.3. Assignment of boundary conditions to finite element mesh

The very nature of the updated front Lagrangian–Eulerian approach to the modelling of free surface flow requires that an automatic method of assigning boundary conditions to the deforming mesh be devised. Boundary conditions assigned to the wall boundary segments are read as input. Upon discretization of the boundary segments into a continuous loop of interconnected line segments, the properties of each line segment, i.e. slope, intercept and type, have to be evaluated. Three types of segments are identified, namely vertical, horizontal and inclined, the vertical and horizontal cases being considered as special cases of the inclined type. The nodes on the metal boundary or interface are then obtained and a check is initiated to find out on which boundary segment the node lies. To achieve this, a loop over the wall segments is performed and the distance of the node to the wall segment is evaluated to identify the contact nodes. The properties of the last segment and the next, corresponding to the boundary node, are evaluated. If the relative difference is less than a set tolerance, with the node deemed to be close enough to the wall, and the wall segment boundary conditions are assigned to the boundary node, Figure 6(i). Here • is a wall boundary node and ○ is the mesh boundary node under consideration.

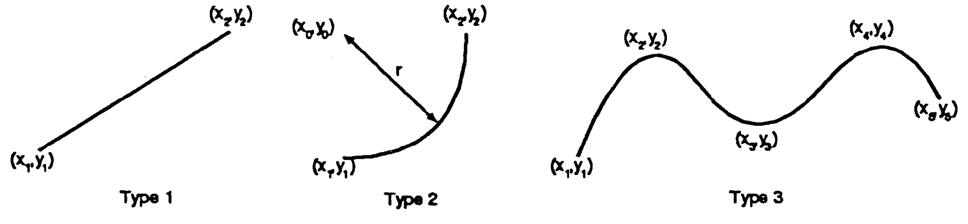


Figure 5. Boundary segment types

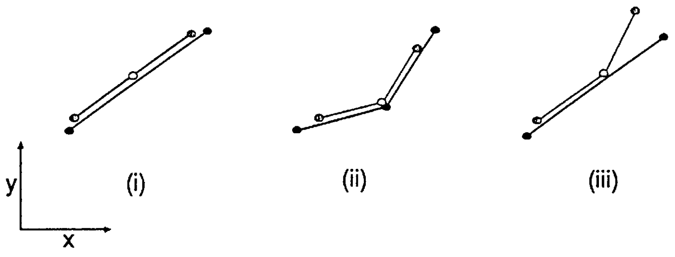


Figure 6. Boundary node types

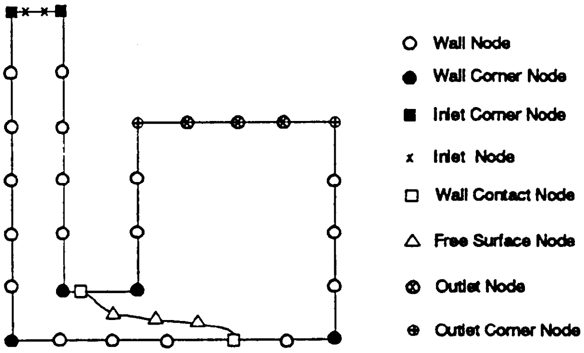


Figure 7. Boundary node types

Nodes that correspond to a global boundary node, Figure 6(ii), are identified by checking the coordinates of the boundary node to see whether it is less than a set tolerance from the global boundary node in question. Should this be the case, boundary conditions are assigned to the node, consistent with input values. Contact nodes, Figure 6(iii), are identified when the slope of the preceding or next segment of a boundary node does not correspond to the slope of the boundary segment on which the node lies. Once these nodes are located, they are flagged for special treatment, such as the magnitude of wall adhesion if modelled, and the corresponding boundary conditions are assigned.

Nodes not deemed to lie on any wall segment are collected and flagged as free surface nodes. The different types of boundary nodes are shown in Figure 7.

6. TIME STEP ADJUSTMENT

6.1. Front-tracking considerations

Within each time step iteration the time taken for the node closest to the boundary to penetrate the wall is computed. If this node is within a set tolerance from the wall, it is fixed to it; otherwise, the time step for the current iteration is modified to prevent penetration. Reduction of the time step is also based on the time taken for the contact node, Figure 7, to move to the end point of a wall segment. This process has to be performed to ensure an element does not violate the boundary integrity. The process is repeated until convergence is attained, with the node deemed not to have penetrated the solid boundary.

When analysing flow in some complex geometries, the percentage of fluid lost can exceed the tolerance set. In this case the time step size is automatically reduced to ensure that the volume of fluid lost does not exceed the acceptable upper limit.

6.2. Time step considerations

In order to automate the time step selection process, the movement of the free surface at any time step was restricted to the neighbouring elements, i.e. the Courant number was limited to 1·0,

$$C = u\Delta t/h \leqslant 1{\cdot}0, \tag{18}$$

where u is the magnitude of the velocity and h is the distance between two corner nodes.

7. FRONT UPDATE

The free surface was traced implicitly by making use of the Lagrangian transport equation applied to the nodes on the front. Velocities obtained at the front were used to increment the free surface via the expression

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \delta\mathbf{x}^n \quad \text{on } \Gamma_\mathrm{f}, \tag{19}$$

with

$$\delta\mathbf{x}^n = \mathbf{v}\delta t, \tag{20}$$

where $\mathbf{x}^{n+1}$ is the nodal position at the $(n+1)$th time step, $\delta\mathbf{x}^n$ is the increment at the nth time step, δt is the current time step, $\mathbf{v}$ is the nodal velocity and Γ_f is the free surface along which the pressure was specified as atmospheric, Figure 8(i). The front was found to be sufficiently accurate when only the free surface corner nodes were incremented, with the midside nodal co-ordinates interpolated.

Upon updating the front nodes, the front elements are checked for boundary violation. For flow around sharp corners an element face can violate the boundary, Figures 8(ii) and 8(iii). Should this happen, the node closest to the boundary node is adjusted to coincide with the boundary nodal position. Another common case of boundary violation is shown in Figure 9. All the boundary nodes ● lie on the wall, but when the finite element mesh is generated, some newly created midside nodes do not lie within the domain. In such cases the co-ordinate of the node violating the boundary integrity is adjusted so as to coincide with the wall or the element is deleted from the finite element mesh if deemed to lie outside the mould cavity. A further check is performed to adjust the co-ordinates of nodes that lie within the boundary layer to lie on the wall.

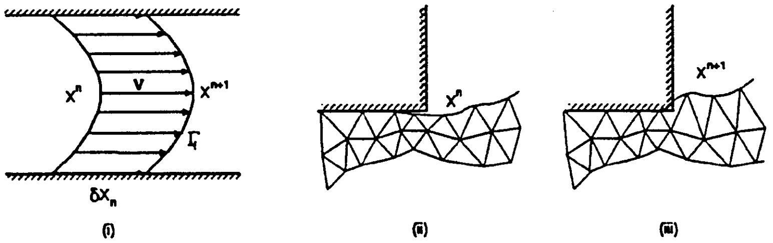


Figure 8. Free surface update

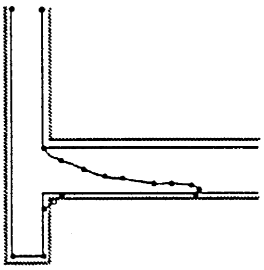


Figure 9. Boundary integrity violation

8. FRONT TRACKING

Upon updating the free surface, the metal boundary data has to be checked and converted to the geometry instructions for the mesh generation module. The following subsections gives an insight into how this is achieved and outlines the checks necessary to ensure an accurate prediction of the free surface movements.

8.1. *Extraction of boundary nodes*

Once the free surface is incremented, the boundary nodes have to be extracted. Boundary nodes are identified by flagging all coincident element faces. Non-coincident faces are boundary faces (Figure 10). These are stored in an anticlockwise order, beginning at the inlet. Owing to memory considerations, only nodes that are critical to the boundary definition, i.e. do not lie on a straight line, are stored.

8.2. *Treatment of air bubbles*

Entrapped air pockets from the previous time step are obtained from the updated mesh. These typically would be an internal enclosed loop of boundary segments. The formation of new air pockets or holes in the incremental mesh is determined by checking the front nodes for overlapping segments, Figure 11(i), or intersecting segments, Figure 11(ii).

This commonly occurs under certain flow conditions, when a wave may develop, resulting in the free surface folding over itself. Should this happen, the volume of the air pocket that results is evaluated. If deemed significant, its effect is included in the mesh geometry file and a hole is generated in the mesh; otherwise, it is ignored. Splashing in the fluid domain, which typically occurs

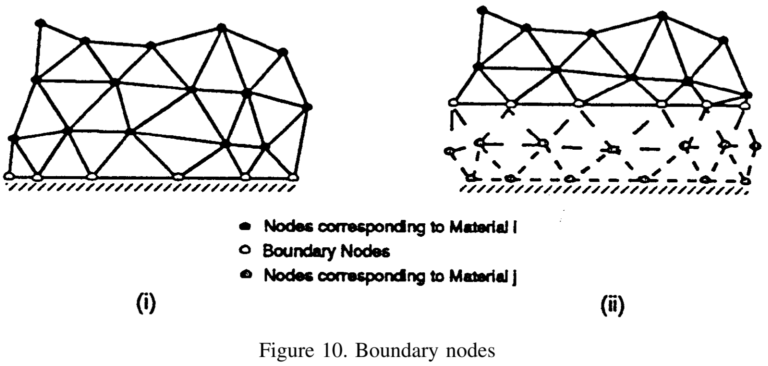

● Nodes corresponding to Material I
○ Boundary Nodes
◐ Nodes corresponding to Material J
(i)
(ii)

Figure 10. Boundary nodes

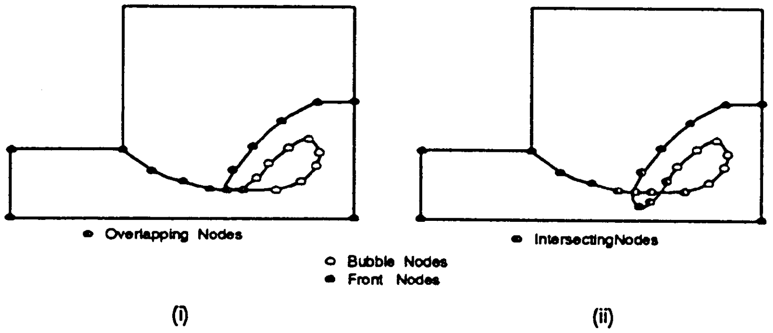


Figure 11. Formation of bubbles

during 'break-up' of the fluid particles away from the main mass of fluid into droplets, is not modelled, Figure 12(i). When this occurs, the co-ordinates of the interesting segments are not modified, to prevent break-up, Figure 12(ii).

8.3. *Flow around obstacles*

While tracking the free surface within a domain with internal obstacles, they are input as segments or subdomains defined by interconnected nodes or lines. As the free surface is updated, all internal subdomains are continuously checked to see whether they are completely enclosed by the moving fluid. Should this be the case, the internal subdomain or segment is identified as a boundary, Figure 13(i), or a hole, Figure 13(ii), and the appropriate boundary conditions applied to the contact nodes.

8.4. *Collision of two moving fronts*

Once convergence has been attained, the front nodes are checked for 'mixing'. Should this happen, the time step is automatically modified to prevent significant loss of fluid. A condition under which this occurs is when the fluid flows in an undular wave-like profile, folding over itself.

8.5. *Spline fitting*

When the free surface has experienced significant distortions, a cubic spline of the form

$$y(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{22}$$

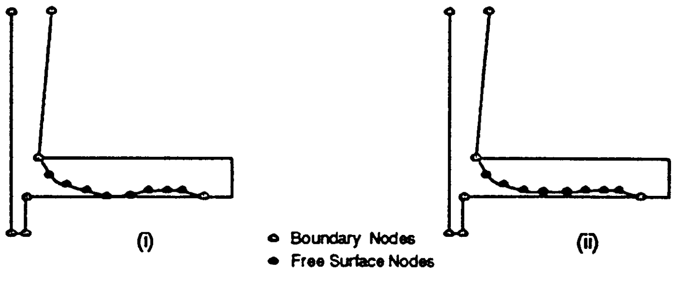


Figure 12. Break-up of free surface

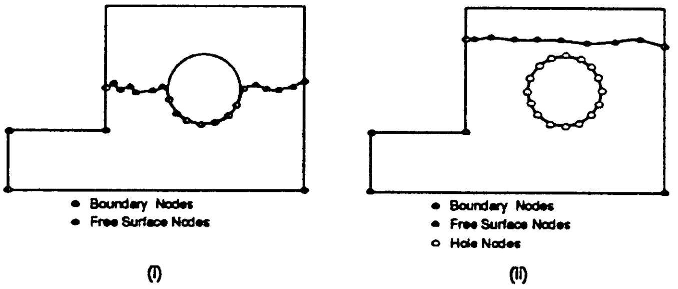


Figure 13. Flow around a circular obstacle

is fitted at intervals along the front where smoothing is required. A composite function is defined through the key points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, with $x_0 < x_1 < \cdots < x_n$. Key points are chosen at specified intervals or in regions of high curvature and the intermediate nodes are computed. In regions of high curvature more nodes are generated to give an accurate definition of the front. $y(x)$ is a cubic polynomial over each subinterval $x_{i-1} \leqslant x \leqslant x_i$, with $i = 1, 2, \ldots, n$. The first and second derivatives are continuous at the data points, resulting in a smooth piecewise curve. Figure 14 shows the result of smoothing a distorted free surface.

8.6. *Contact angle check*

A minimum contact angle has to be specified for the contact node. As the fluid flows, it sometimes tends to spread along the retaining wall. As this happens, the contact angle might tend to 0°, making it difficult to regenerate the mesh. Currently the minimum contact angle allowed is 5°, Figure 15. The

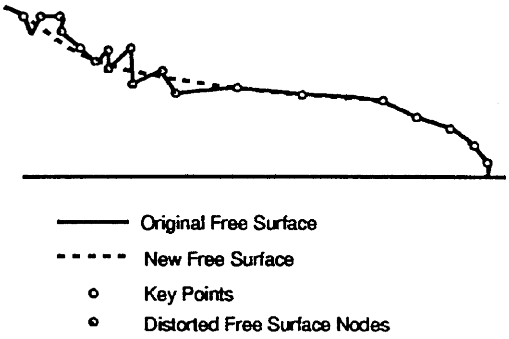


Figure 14. Modified free surface using cubic spline

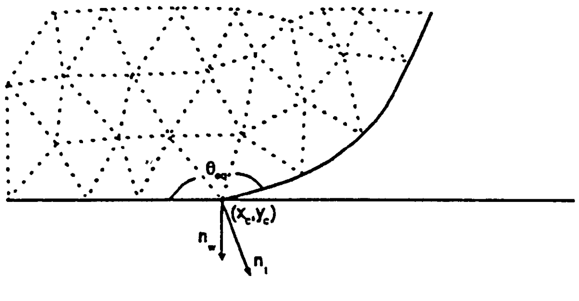


Figure 15. Contact point normal

choice is based on geometrical considerations, although when modelling wall adhesion, it is an input boundary condition which depends on the fluid and wall conditions.

9. MESH GENERATION

An unstructured mesh is generated using a Delaunay triangulation algorithm. A domain composing the specified geometry is spatially discretized using an octree subdivision.[36]

The triangulation process is performed over the whole geometry, making it necessary to delete triangles that are located outside the subdomain in question. The directionality of the boundary edges and of each triangle is related to the corresponding area. If the area is negative, the triangle is considered to lie outside the subdomain and is deleted.

Two methods are used to deal with missing boundaries. Either temporary points could be inserted for every missing edge, to aid in the element rejection procedure, or the boundaries could be directly checked and recovered. This would normally result in the rejection of some elements and the addition of new ones.

9.1. *Mesh quality improvement*

In order to have an accurate prediction of the front movements, the finite element mesh was refined along the free surface. Three mesh density control parameters are currently used, one for the free surface, another for the solid wall boundary and the third for the interior fluid domain. The specified mesh density is then interpolated to allow for cell division using the expression

$$d_{\mathrm{p}} = \sum_{i=1}^{n} \frac{d_i}{r_i^{\alpha}} \Bigg/ \sum_{i=1}^{n} \frac{1}{r_i^{\alpha}} \tag{22}$$

for $1{\cdot}0 \leqslant \alpha \leqslant 2{\cdot}0$, where n is the number of control points and r^i is the distance between the current point and the ith control point. Currently only the control points in the neighbourhood of the point in question are selected to ensure that the local features are reflected.

By discretization of the boundary information into a number of interconnected boundary nodes, some control over the front node spacing can be obtained, especially when the density specified is greater than the node spacing.

A Laplacian smoothing technique is used to smooth out the mesh, thereby prolonging its useful life. In this method the position X_i of an interior node i is repositioned based on the expression

$$X_i = \frac{1}{n} \sum_{j=1}^{n_i} X_j \tag{23}$$

for $i = 1, \ldots, m$, where n_i is the number of nodes connected to node i, X_j is the position of the connected nodes and m is the total number of interior nodal points. The result of the above expression is to place the ith node at the centroid of the connected nodes. It has been found for current applications that three smoothing iterations suffice to give a better quality mesh.

Subsequent to the smoothing operation the new co-ordinate of the adjusted node is checked to ensure it does not move outside the boundary or result in an element having a negative area. In the event of any of the above happening, the expression

$$X_i^{k+1} = X_i^k + \frac{\varphi}{n_i} \sum_{j=1}^{n_i} (X_j^k - X_i^k) \tag{24}$$

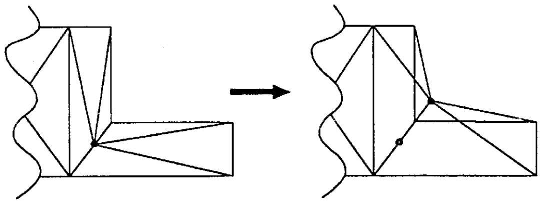


Figure 16. Case of failure of Laplacian smoothing technique

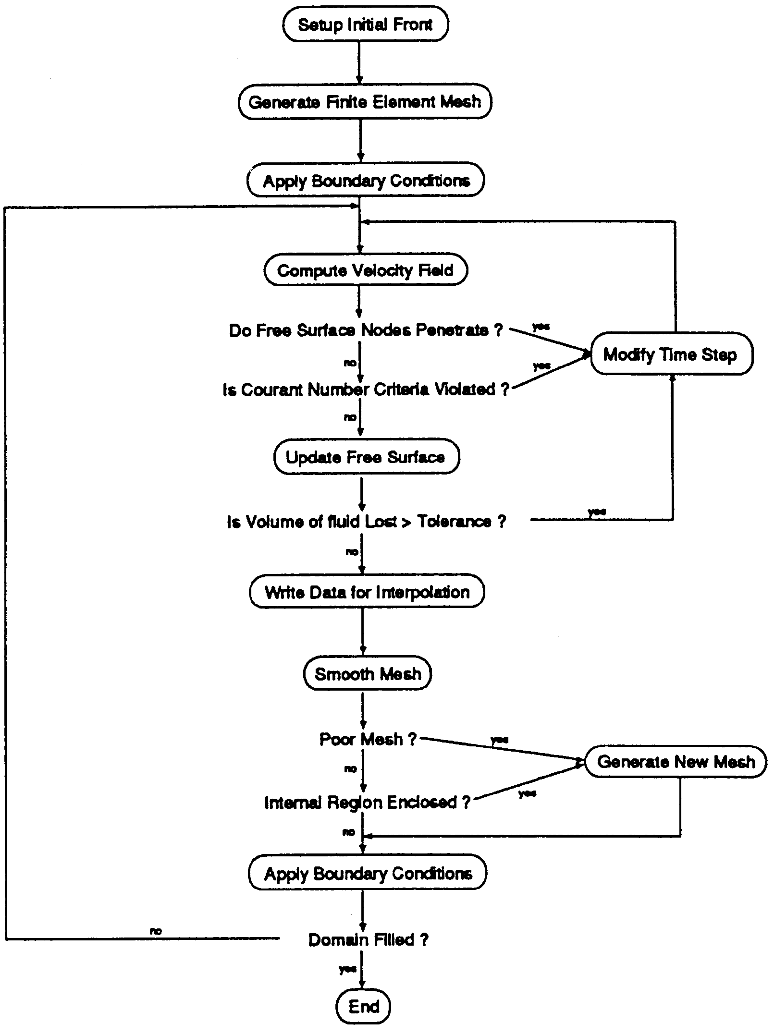

Setup Initial Front
Generate Finite Element Mesh
Apply Boundary Conditions
Compute Velocity Field
Do Free Surface Nodes Penetrate ?
yes
no
yes
Is Courant Number Criteria Violated ?
Modify Time Step
no
Update Free Surface
Is Volume of fluid Lost > Tolerance ?
yes
no
Write Data for Interpolation
Smooth Mesh
Poor Mesh ?
yes
no
Generate New Mesh
Internal Region Enclosed ?
yes
no
Apply Boundary Conditions
no
Domain Filled ?
yes
End

Figure 17. Flow chart

is used, where $i = 1, \ldots, m$, the subscript k denotes the iteration number and φ is a relaxation parameter. The above expression, while being a variation of the first smoothing equation, does not guarantee that an interior node will not move outside the geometry (Figure 16). All the checks stated above have to be performed again to establish that the adjustments do not result in a bad mesh.

10. TRANSPORT OF STATE VARIABLES

Upon regeneration of the mesh, state variables are interpolated from the old mesh to the new one. The location of newly generated points on the updated mesh from the previous time step is found. Using area co-ordinates of the old element in which they lie, the values of the primitive variables at the corner nodes are interpolated onto the new nodes. A flow chart for the complete filling algorithm is shown in Figure 17.

11. MODELLING OF TILT POURING

The modelling of tilt pouring, a process commonly utilized in foundaries to reduce the effect of turbulence, has been implemented using a correlation between the percentage metal in the mould and the tilting angle.

The input parameters required are the minimum and maximum tilt angles and the percentages of fluid in the mould below which a minimum tilting angle is required and above which the maximum tilting angle is used. The gravitational body force $\mathbf{f}$ in equation (2) is then evaluated using the expressions

$$f_x = -(g_x \sin\theta + g_y \cos\theta), \tag{25}$$

$$f_y = (g_x \cos\theta + g_y \sin\theta), \tag{26}$$

where g_x and g_y are the input body forces and θ is the tilting angle measured positive anticlockwise.

The value of θ is automatically derived from the expression

$$\theta = \theta_{\min} + (\theta_{\max} - \theta_{\min}) \times \frac{\%_{\text{full}} - \%_{\min}}{\%_{\max} - \%_{\min}}, \tag{27}$$

where $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum tilting angles and $\%_{\text{full}}$, $\%_{\min}$ and $\%_{\max}$ are the percentage of fluid in the mould, the percentage of fluid below which a minimum tilting angle is desired and the percentage of fluid above which the maximum tilting angle is used.

12. NUMERICAL EXAMPLES

12.1. Dam break analysis

A broken dam problem (Figure 18) for which experimental data were available was used to validate the model. The data available were presented in the form of a non-dimensionalized plot of the distance of the leading edge from the bottom left-hand corner versus time.[24,37] In order to simulate the dam-breaking process, a rectangular column of water 0·05715 m wide and 0·05715 m high in hydrostatic equilibrium was allowed to collapse under gravity in a 0·17145 m wide and 0·07 m high rectangular cavity as the right-hand side wall is removed. Free slip boundary conditions were implemented at the solid wall boundaries, with the gravitational acceleration of 9·81 m s^{-2} acting vertically downwards. The density of water is 1000·0 kg m^{-3} and the viscosity 0·5 kg m^{-1} s. The

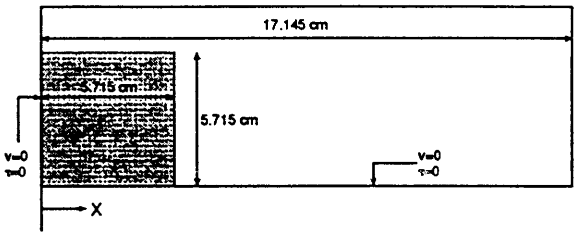


Figure 18. Schematic diagram of broken dam problem

time step used for the analysis was of the order of 0·005 s. The tolerance for the iteration was 1·0%, with nodes within 0·0001 m from the wall assumed to lie on it.

Figure 19 shows a comparison of the numerical results obtained against experimentally derived values. For convenience the dimensionless time and front positions are defined as $t\sqrt{(g/H)}$ and x/H respectively. The results obtained are generally within 5% of the published experimental data.

Figure 20 shows the free surface position and mesh at different time steps. It can be seen that the free surface has a well-behaved undular shape.

The analysis took 28 time steps to run, with an average iteration time of approximately 3 min, on a SUN Sparc 10 workstation. The local percentage of volume of fluid lost was 3·0%. In this analysis the mesh was regenerated at every time step, with a lot of I/O instructions. Restricting the number of remeshing operations and I/O operations results in significant speed-up of the analysis time.

12.2. Filling a bicycle hub

The filling of an industrial casting was modelled numerically using the algorithm described above to illustrate the effectiveness of the method implemented in tracking free surfaces. Two sets of results are presented, one without tilting and the other with tilting.

The metal was cast in a sand mould and filled from the top. Figure 21 shows the geometry of the mould cavity, the initial metal front and mesh. The density of aluminium is 2385·0 kg m^{-3}, with a gravitational acceleration of 9·8 m s^{-2}. The dynamic viscosity used was 1·14695 kg m^{-1} s. The tolerance for iterations was set at 1·0%, with nodes within a tolerance of 0·0001 m from the wall assumed to lie on it. A time step of between 0·005 and 0·001 s was used.

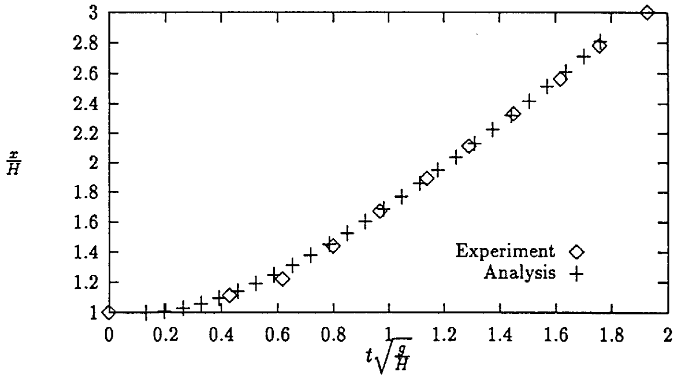
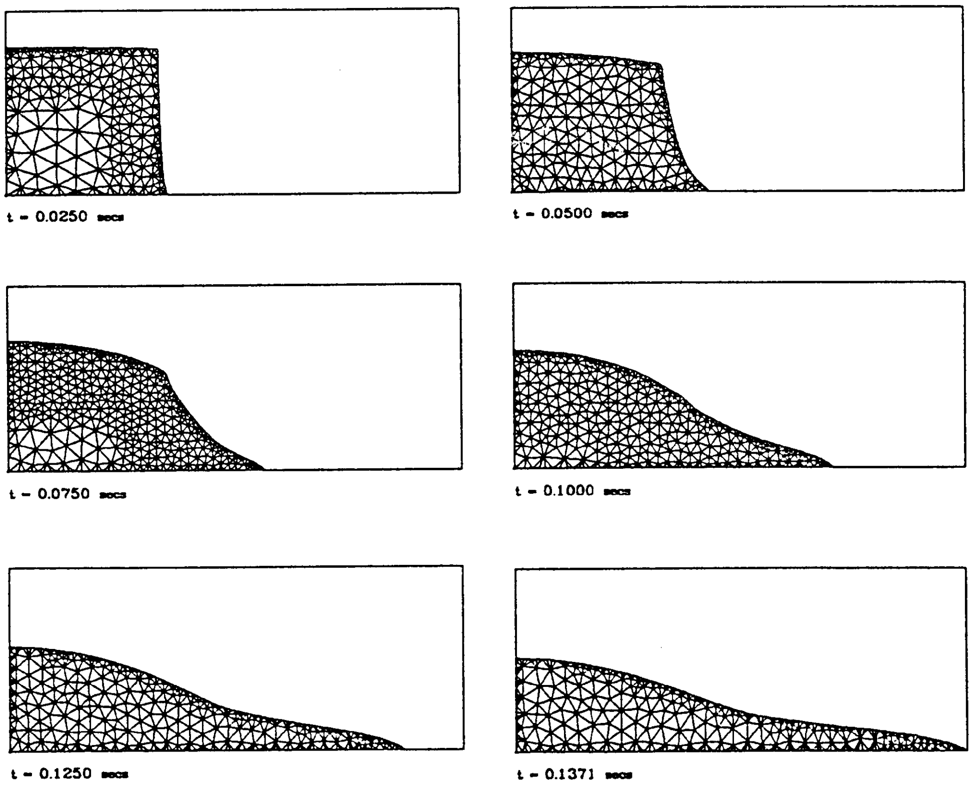


Figure 19. Front position versus time


t = 0.0250 secs

t = 0.0500 secs

t = 0.0750 secs

t = 0.1000 secs

t = 0.1250 secs

t = 0.1371 secs

Figure 20. Mesh and free surface plots for broken dam problem

In these examples the mesh was once more generated at every time step. The mesh density varied linearly within a set distance from the free surface. Outside this region a surface mesh density was imposed within the domain and a wall density along the wall.

As the analysis progressed, these initial values were automatically adjusted to satisfy the stability requirements. The time step could also be automatically adjusted, within limits, should the front

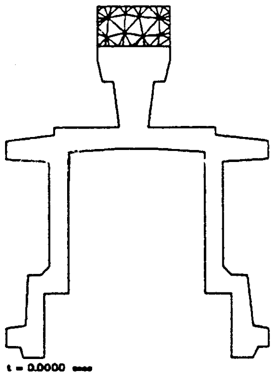


Figure 21. Schematic diagram of bicycle hub

movement be considered too large or to minimize fluid loss. The grid Reynolds number was also continuously monitored. In the event of it exceeding its threshold, 25 for this example, the mesh is automatically refined. Other situations in which refinement is performed are when a zero pivot is encountered or when the free surface is complicated. In the case of a zero pivot being encountered, the location is identified and the mesh is locally refined.

Figure 22 shows the meshes and free surface positions obtained for the analysis without tilting. For the analysis with tilting the tilt angle was maintained at 10° when the mould was less than 10% full and increased uniformly thereafter to a maximum of 90° (i.e. vertical) when more than 90% full. The results obtained are shown in Figure 23.

The results presented illustrate the ability of the method adopted to capture complex free surface patterns. The finer the mesh, the more accurate are the results obtained. However, owing to run time considerations, care has to be taken in selecting an optimal mesh density control parameter.

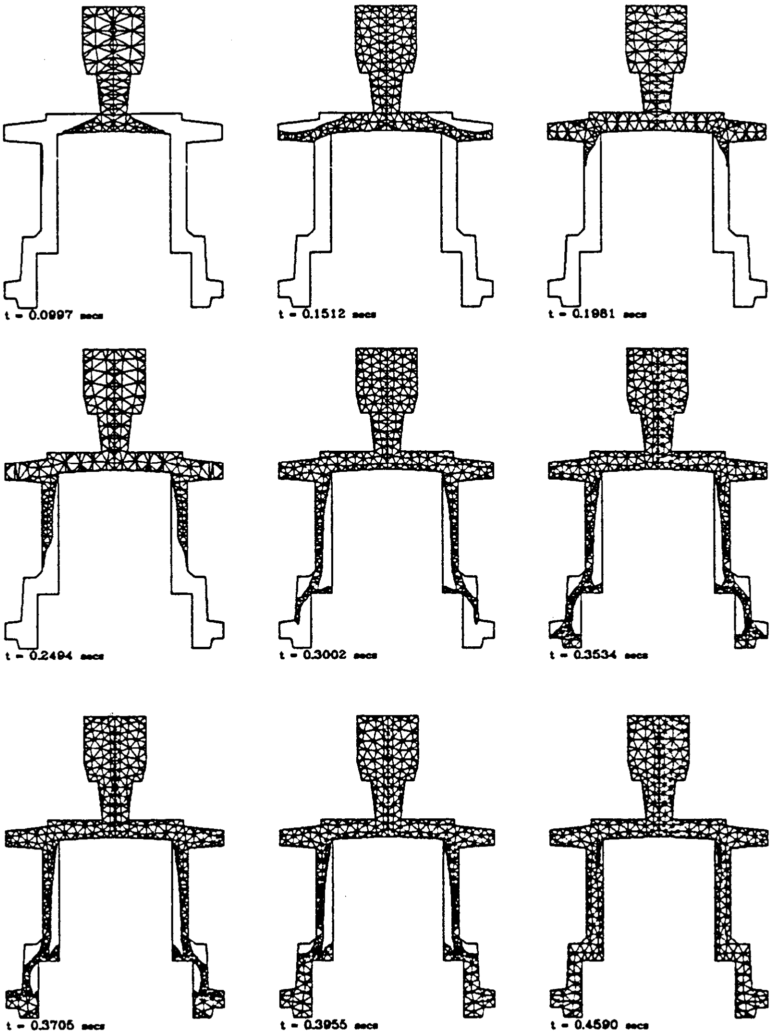


Figure 22. Front position and mesh plots for analysis with no tilting

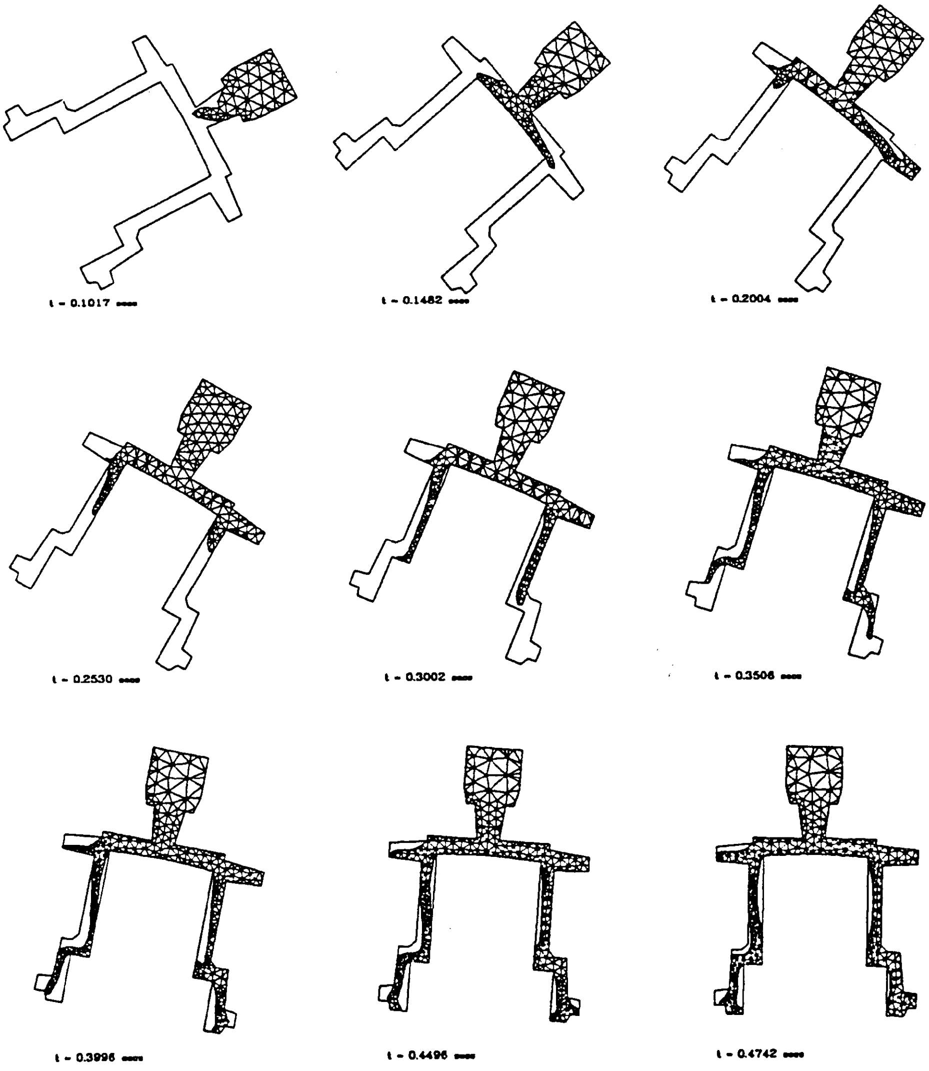


Figure 23. Front position and mesh plots for analysis with tilting

13. CONCLUSIONS

The effectiveness of an updated free surface Lagrangian–Eulerian finite element kinematic description in simulating free surface flow problems, particularly mould filling, has been demonstrated. A mixed interpolation formulation has been successfully used to approximate the discretized governing equations for elimination on a Lagrangian-type moving mesh. Although requiring additional memory requirements and computations when dealing with front tracking as compared with other fixed mesh approaches, significant savings in CPU time are realized by virtue of the air domain not being considered in the finite element analysis. Unlike with the VOF method, an explicit determination of the free surface results, making it ideal for the application of free surface boundary conditions. Difficulties associated with tracking the free surface will be significantly reduced if the above method is applied to the modelling of engineering processes where the free surface experiences less distortion. Results obtained are in good agreement with experimentally derived data.

REFERENCES

1. J. Donea, 'Arbitrary Lagrangian–Eulerian finite element methods', in T. Belytchko and T. J. R. Hughes (eds), *Computer Methods for Transient Analyses*, Elsevier, Amsterdam, 1983, pp. 474–516.
2. C. W. Hirt, A. A. Amsden and J. L. Cook, 'An arbitrary Lagrangian–Eulerian computing method for all flow speeds', *J. Comput. Phys.*, **14**, 227–253 (1974).
3. B. Ramaswamy and M. Kawahara, 'Arbitrary Lagrangian–Eulerian finite element method for unsteady, convective, incompressible viscous free surface flow', *Int. j. numer. methods fluids*, **7**, 1053–1074 (1987).
4. F. Muttin, T. Coupez, M. Bellet and J. L. C. Chenot, 'Lagrangian finite element analysis of time dependent viscous free surface flow using an automatic remeshing technique: applied to casting flow', *Int. j. numer. methods eng.*, **36**, 2001–20015 (1993).
5. B. Debbaut, 'Numerical simulation of the blow molding process', *SPE Antec Papers*, **32**, 1870–1872 (1993).
6. H. S. Kheshgi and L. E. Scriven, 'Penalty finite element analysis of time dependent two dimensional free surface film flow', in T. Kawai (ed.), *Proc. 4th Int. Symp. on Finite Element Methods in Flow Problems*, University of Tokyo Press, Tokyo, 1982, pp. 113–120.
7. F. H. Harlow and J. E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface', *Int. Phys. Fluids*, **8**, 2182–2189 (1965).
8. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, 'The MAC method', *Los Alamos Scientific Laboratory Rep. LA-3425*, 1965.
9. S. P. Wang and K. K. Wang, 'A net inflow method for incompressible viscous flow with moving free surface', *Int. j. numer. methods heat fluid flow*, **18**, 669–694 (1994).
10. G. P. Backer, 'Finite element free surface flow analysis: a new tool for foundary engineers', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 405–412.
11. E. Broyer, C. Gutfinger and Z. Tadmore, 'A theoretical model for cavity filling process in injection molding', *Trans. Soc. Rheol.*, **19**, 423–444 (1975).
12. M. Gao, G. Dhatt and A. B. Cheikh, 'A finite element simulation of metal flow in moulds', *Proc. Sixth Int. Conf. for Numerical Methods in Thermal Problems*, Pineridge Press, Swansea, July 1989.
13. J. R. Hartin, 'A comparison of implicit and explicit solution techniques and results in numerical casting simulations', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Process VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 373–380.
14. G. Kreziak, P. Gilotte and R. Hamar, 'Modelling of fluid flow during mould filling', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 435–442.
15. H. J. Lin and H. L. Tsai, 'Numerical simulation of an integrated filling–solidification casting system', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 380–388.
16. D. M. Lipinski, W. Schaefer and E. Flender, 'Numerical modelling of the filling sequence and solidification of castings', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 389–396.
17. B. Minaie, K. A. Stelson and V. R. Voller, 'Fluid flow and solidification model of die casting', *ASME Model. Mater. Process.*, **3**, 35–50 (1987).
18. I. Ohnaka, 'Modelling of fluid flow and solidification in casting', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 337–348.
19. C. R. Swaminathan and V. Voller, 'Numerical modelling of filling and solidification in metal casting processes; a unified approach', in R. W. Lewis (eds.), *Proc. Int. Conf. for Numerical Methods in Thermal Problems VIII*, Pineridge Press, Swansea, July 1993, pp. 284–296.
20. M. R. Tadayon, J. A. Spittle and S. G. R. Brown, 'Fluid flow and heat transfer modelling of mould filling in casting processes', in R. W. Lewis (eds.), *Proc. Int. Conf. for Numerical Methods in Thermal Problems VIII*, Swansea, July 1993, pp. 309–317.
21. A. S. Usmani, J. T. Cross and R. W. Lewis, 'A finite element model for the simulations of mould filling in metal casting and associated heat transfer', *Int. j. numer. methods eng.*, **35**, 787–806 (1992).
22. D. M. Waite and T. M. Sammonds, 'Finite element free surface modelling', in T. S. Piwonka, V. Voller and L. Kagermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 357–364.
23. Y. F. Zhang and W. K. Liu, 'Casting filling simulation of thin-walled cavities with solidification', in T. S. Piwonka, V. Voller and L. Katgermann (eds), *Modelling of Casting, Welding and Advanced Solidification Processes VI*, Minerals, Metals and Materials Society, Warrendale, Pennsylvania, 1993, pp. 413–420.

24. C. W. Hirt and B. D. Nichols, 'Volume of fluid VOF method for the dynamics of free surface boundaries'. *J. Comput. Phys.*, **39**, 210–225 (1981).
25. W. S. Hwang and R. A Stoehr, 'Modelling of fluid flow', in *ASM Metals Amsterdam, Handbook*, Vol. 15, 9th edn, ASM, Metals Park, OH, 1988, p. 867.
26. M. S. Engelman, J. L. Sani and P. M. Gresho, 'The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow', *Int. j. numer. methods fluids*, **2**, 225–238 (1982).
27. G. F. Pinder and W. G. Gray, *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic, New York, 1977.
28. D. S. Burnett, *Finite Element Analysis—From Concept to Applications*, Addision-Wesley, Reading, MA, 1987.
29. T. J. R. Hughes, *The Finite Element Method—Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliff, NJ, 1987.
30. O. C. Zienkiewicz, *Programming the Finite Element Method*, McGraw-Hill, New York, 1977.
31. C. Taylor and T. G. Hughes, *Finite Element Programming of the Navier Stokes Equations*, Pineridge, Swansea, 1981.
32. O. C. Zienkiewicz, R. L. Taylor and J. A. Baynham, 'Mixed and irreducible formulations in finite element analysis', in S. N. Atluri, R. H. Gallagher and O. C. Zienkiewicz (eds), *Hybrid and Mixed Finite Element Method*, pp. 405–431, Wiley, Chichester, 1977.
33. O. C. Zienkiewicz, Y. C. Liu and G. C. Huang, 'Error estimates and convergence rates for various incompressible elements', *Int. j. numer. methods eng.*, **28**, 2191–2202 (1989).
34. T. J. R. Hughes, W. K. Liu and T. K. Zimmermann, 'Lagrangian–Eulerian finite element formulations for incompressible flows', *Comput. Methods Appl. Mech. Eng.*, **29**, 329–349 (1981).
35. T. J. R. Hughes, 'The analysis of transient algorithms with particular reference to stability behaviour', in *Computer. Methods for Transient Analysis*, pp. 67–155, Elsevier, Amsterdam, 1989.
36. Y. Zheng, 'Three dimensional mesh generation and visualisation for finite elements: principle and practice', *Ph.D. Thesis C/PhD/183/1994*, University of College of Swansea, 1994.
37. J. C. Martin and W. J. Moyce, 'An experimental study of the collapse of liquid columns on a rigid horizontal plane', *Phil. Trans. R. Soc. Lond. A*, **244**, 312–324 (1952).
38. R. Codina, U. Schäfer and E. Onãte, 'Mould filling simulation using fintie elements', *Int. j. numer. methods. heat fluid flow*, **4**, 291–310 (1994).