INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 407–420 (1997)

A DEFORMING FINITE ELEMENT MESH FOR USE IN MOVING ONE-DIMENSIONAL BOUNDARY WAVE PROBLEMS

C. D. CHRISTIAN[1]* AND G. N. PALMER[2]

[1] *Department of Civil and Resource Engineering, University of Auckland, Auckland, New Zealand*
[2] *Environment Waikato, Hamilton East, New Zealand*

SUMMARY

The finite element method is developed to solve the problem of wave run-up on a mild, plane slope. A novel approach to implementing a deforming mesh of one-dimensional, three-node, isoparametric elements is described and demonstrated. The discrete time interval (DTI), arbitrary Lagrangian–Eulerian (ALE) and space–time element (STE) methods are used to solve the unsteady one-dimensional shallow water wave equations. The boundary condition required is simply the seaward water surface elevation, and although the method has only been tested for monochromatic waves, it should be equally valid for any sea state which can be described as a water surface elevation as a function of time. All three solution methods are shown to given good results. Time histories of the terms of the governing equations are calculated and used to demonstrate how the ALE and STE methods account for mesh deformation. The model could be extended to two dimensions, which would have practical application to the run-up of obliquely incident waves. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 407–420 (1997).

No. of Figures: 8. No. of Tables: 3 No. of References: 11.

KEY WORDS: wave; finite element; deforming mesh; run-up

1. INTRODUCTION

Numerical modelling of wave run-up involves boundary movement as the computation advances in time. The calculation procedure is typically

(i) calculate the position of the moving boundary (water surface and/or run-up tip)
(ii) discretize the spatial domain of the region being modelled (finite difference grid or finite element mesh) and
(iii) solve the equations governing flow within the region, typically one or two dimensions, defined by the moving boundary.

Since the position of the boundary is dependent on the near-field flow regime, which in turn is dependent on the former, an iterative calculation procedure is required at each time step.

Many models are characterized by a fixed one-dimensional finite difference grid on which the unsteady shallow water wave equations are solved. These equations are based on the assumptions that the slope is mild and impermeable and the water pressure is hydrostatic. Kobayashi *et al.*,[1] Allsop *et al.*[2] and Van der Meer and Breteler[3] have demonstrated good results with this approach, even for

* Correspondence to: C. D. Christian, Department of Civil and Resource Engineering, University of Auckland, Auckland, New Zealand.

CCC 0271–2091/97/040407–14 $17.50

Received March 1995
Revised August 1996

relatively steep slopes. They define a grid larger than the maximum expected size of the spatial domain before computations start and then solve the governing equations at the nodes that are 'wet' during the time step under consideration. A Eulerian reference system is used which is comparable with that normally used for fixed boundary problems.

One drawback of this approach is that the grid cannot be refined at locations of interest, such as the run-up tip, without refining it everywhere else. Further, the Courant stability criterion dictates that the run-up tip must not move a distance greater than one grid interval during each time step. This requires a high grid resolution in both the space and time domains in order to model its position accurately.

A deforming mesh of finite elements can overcome these limitations, largely because of its inherent co-ordinate transformation capability. Gopalakrishnan and Tung[4] describe one such approach in which nodes and elements are added to the mesh as the run-up tip moves along the slope. Only uprush is modelled, apparently because their model cannot deal with the removal of nodes which would be necessary during downrush.

This paper describes a novel approach to implementing a deforming finite element mesh in which the number of elements within the mesh remains constant. One advantage is that the global element matrix does not need to be restructured at each time step, making the solution method more efficient. A node is positioned at the run-up tip and allowed to move with it. At the same time the mesh is allowed to deform and some or all internal nodes move, termed rezoning.

The discrete time interval (DTI), arbitrary Lagrangian–Eulerian (ALE) and space–time element (STE) methods are used to solve the unsteady one-dimensional shallow water wave equations. Each method treats the solution as an initial value problem at each time step. The mesh-rezoning scheme is demonstrated by simulating wave run-up on a smooth, mild slope. The manner in which the governing equations account for mesh deformation is discussed by calculating and comparing time series of the terms of the continuity equation. The work forms part of an attempt to model wave run-up on the seaward face of a rubble mound breakwater.[8]

2. GOVERNING EQUATIONS

The problem being modelled is shown in Figure 1. It is assumed that the slope is smooth and impermeable, the fluid pressure is hydrostatic and the vertical fluid particle accelerations equal zero. In accordance with these assumptions the equations to be solved are the unsteady one-dimensional shallow water wave equations. These consist of the continuity equation

$$\frac{\partial}{\partial t}(h+\eta)+U\frac{\partial}{\partial x}(h+\eta)+(h+\eta)\frac{\partial U}{\partial x}=0 \tag{1}$$

and the momentum equation

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+g\frac{\partial \eta}{\partial x}=0, \tag{2}$$

where η is the water surface deviation from the mean level, U is the depth-averaged horizontal velocity and g is the acceleration due to gravity.

The boundary conditions are

$$\eta=f(t) \tag{3}$$

at the seaward boundary and

$$U=U_{\mathrm{R}} \tag{4}$$

at the run-up tip, where U_{R} is the horizontal velocity of the run-up tip.

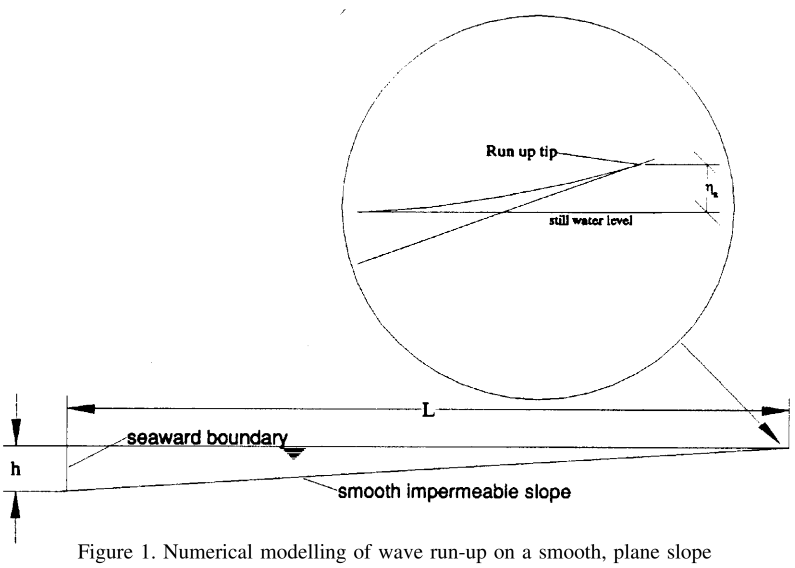

Figure 1. Numerical modelling of wave run-up on a smooth, plane slope

3. BOUNDARY MOVEMENT

An iterative procedure is used at each time step in which

$$U_R = U_R^i \tag{5}$$

at the first iteration at $t = t^{i+1}$ and

$$U_R = \mathrm{d}X_R/\mathrm{d}t \tag{6}$$

with

$$X_R = f(\eta_R) \tag{7}$$

at subsequent iterations, where η_R is the elevation of the run-up tip (Figure 1). Equation (6) is solved numerically using the time-marching θ-method[9] and equation (7) is a kinematic condition requiring the run-up tip to move along the surface of the slope. Iterations continue at each time step until satisfactory convergence is achieved for the position of the run-up tip.

4. MESH REZONING

4.1. Node movement

Computations begin by discretizing the region between the seaward boundary and run-up tip with a mesh of one-dimensional finite elements. The length of the spatial domain changes at subsequent time steps as a result of moving the node at the run-up tip. Some or all internal nodes are also moved, distributing the change in length over some or all elements. This part of the mesh is referred to as the variable length zone.

In selecting a mesh-rezoning scheme for this type of model, the effect that node movement has on related computations must be considered. For example, global derivatives and the size of the domain of integration must be recomputed for an element every time the nodes directly connected to it move. To minimize computational time, nodes should be moved only when necessary.

Two types of criterion are generally available for mesh rezoning and both are used in the model described here. The first treats nodes as Lagrangian markers which track individual fluid particles. This criterion is applied to the node at the run-up tip along with other user-selected nodes. The second uses a geometrical criterion with the objectives of reducing relative element distortion and retaining locally refined portions of the mesh. This is generally easier to satisfy than the former criterion because some tolerance on node position is permitted. A second advantage is that node movement, and hence computational time, is minimized.

4.2. Node types

To facilitate mesh rezoning, each node is classified before computations begin as one of the four types listed in Table I. This classification identifies the nodes which are permitted to move and remains constant during computations. The particular combination of node types also controls how the mesh deforms.

A type E1 node is permanently fixed in position and maintains the horizontal co-ordinate that was calculated or specified as part of the initial conditions. The node at the seaward boundary must be specified as an E1 node because the measured or calculated time series data used as the boundary condition are defined at fixed points in space.

For each of the type E2 nodes the *potential* horizontal distance moved by the *j*th global node, δ_j, is computed by linear interpolation within the mesh. The model automatically determines which of the type E2 nodes *actually* move at each time step by comparing the value of δ_j with what is termed here the node position tolerance δ. The value of δ is specified by the user before computations begin and is constant both spatially and temporally. As depicted in Figure 2, if $|\delta_j| \geqslant \delta$, then the node is moved a distance δ_j. This avoids node movement that would increase computational time without significantly improving the distribution of nodes. The use of δ is equivalent to placing a tolerance on node position.

In most situations it is desirable to have the mesh finely discretized in the vicinity of the run-up tip so that its motion is modelled accurately. This can be achieved by defining some nodes adjacent to the run-up tip as type E3 nodes which move the same distance and in the same direction as the node at the run-up tip. Mesh deformation is then accommodated elsewhere, permitting a time-invariant node spacing to be retained next to the run-up tip. This is an improvement on the mesh-rezoning schemes used by Lynch and Gray[6] and Gopalakrishnan and Tung[4] in which the node spacing next to the run-up varied with time.

Type E4 nodes are Lagrangian markers attached to fluid particles. Their co-ordinates are calculated by integrating

$$\mathrm{d}X/\mathrm{d}t = U \tag{8}$$

using the θ-method.

Table I. Definition of node types and rezone codes

Node type	Description	Rezone code
E1	Fixed in position	0
E2	Movement of *j*th global node permitted subject to following criteria:	
	if $|\delta_j| < \delta$, then retain current position	2
	if $|\delta_j| \geqslant \delta$, then move node	1
E3	Moved same distance and in same direction as node at run-up tip	1
E4	Lagrangian node	1

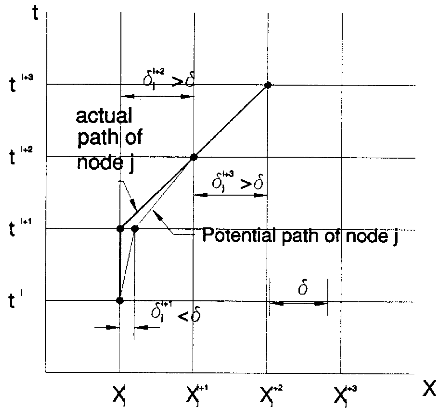

Figure 2. Effect of δ on movement of type E2 nodes

The four node types give the user considerable choice in how the mesh deforms. If, however, the mesh is too coarse at a location where the velocity gradient is high, then node spacings could inadvertently reduce to zero. This error condition is checked at each time step immediately after mesh rezoning and, if detected, computations are automatically terminated. It can be avoided in subsequent simulations by reducing the value of δ, refining the mesh or introducing a set of type E3 nodes.

4.3. *Recalculation of co-ordinate-dependent variables*

To minimize recalculation of co-ordinate-dependent variables such as global co-ordinates, nodal co-ordinates are compared at adjacent time steps. If they differ, then this indicates that the element has deformed since the last time the recalculations were performed and hence must be repeated.

If the DTI method is being used, then previous time step values of η and U must also be interpolated (Figure 3). To facilitate this, the rezone codes shown in Table I are defined. These codes are both assigned and updated automatically during computations. A rezone code equal to 0 is associated with the type E1 nodes which are permanently fixed in position. This indicates that interpolation is not required since the nodes do not move. In comparison, any node that is moved is assigned a rezone code equal to 1, indicating that interpolation is required at that node. A rezone code equal to 2 indicates that although the node is a type E2 node, it has not been moved during the previous time step or iteration. Consequently, interpolation is not necessary. During computations the rezone code of a type E2 node will therefore alternate between values of 1 and 2.

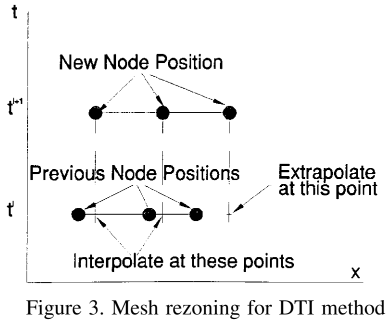

Figure 3. Mesh rezoning for DTI method

5. SOLUTION METHODS

The Galerkin weighted residual method[9] is used to solve (1) and (2). These are written in weighted residual form for an element as

$$\int_{\Omega}\left(\omega\frac{\partial}{\partial t}(h+\eta)+\omega U\frac{\partial}{\partial x}(h+\eta)+\omega(h+\eta)\frac{\partial U}{\partial x}\right)\mathrm{d}\Omega=0, \tag{9}$$

$$\int_{\Omega}\left(\omega\frac{\partial U}{\partial t}+\omega U\frac{\partial U}{\partial x}+\omega g\frac{\partial\eta}{\partial x}\right)\mathrm{d}\Omega=0, \tag{10}$$

where ω is a weighting function and Ω is the domain of integration. The Bubnov–Galerkin method is used in which the weighting functions are the same type as the basis (shape) functions.

Quadratic basis functions with c^0 continuity are used in the spatial domain along with three-point Gauss–Legendre numerical integration. In addition, the STE method uses linear basis functions with the two-point Gauss–Legendre integration in the time domain.

5.1. DTI method

The distinguishing feature of the DTI method is that previous time step values of η and U are interpolated at each time step and then computations continue as an initial value problem (Figure 3).

Basis functions are functions of the co-ordinate x and the nodal values are functions of time t. Global derivatives are therefore of the form

$$\frac{\partial\eta(x,t)}{\partial x}=\frac{\mathrm{d}\psi(x)}{\mathrm{d}x}\eta(t), \tag{11}$$

$$\frac{\partial\eta(x,t)}{\partial t}=\psi(x)\frac{\mathrm{d}\eta(t)}{\mathrm{d}t}. \tag{12}$$

A one-dimensional isoparametric element is defined in which the local co-ordinate ξ corresponds to the global co-ordinate x.

The time derivatives are evaluated by a finite difference scheme based on an uncentred implicit formulation. The solution is advanced in time an amount Δt, with the weighted residuals formed within the spatial domain at $t=t^i+\theta\Delta t$, where θ has a value between 0 and 1.

Nodal values such as $\eta(t)$ are evaluated using

$$\eta(t)=\eta^{\theta}=(1-\theta)\eta^*+\theta\eta^{i+1}, \tag{13}$$

where η^* is the vector of interpolated nodal values at $t=t^i$ and η^{i+1} is the vector of nodal unknowns whose values are sought (at $t=t^{i+1}$). Global time derivatives are approximated by

$$\frac{\partial\eta(t)}{\partial t}=\frac{\eta^{i+1}-\eta^*}{\Delta t}. \tag{14}$$

The global set of non-linear algebraic element equations is linearized using the Newton–Raphson method.[9] Iterations are performed simultaneously with those used to solve (6). The global system of algebraic equations

$$S_{\mathrm{g}}^{k}\Delta\phi_{\mathrm{g}}^{k+1}=-f_{\mathrm{g}}^{k} \tag{15}$$

is then solved for the vector of corrections to the global nodal values, $\Delta\phi_{\mathrm{g}}^{k+1}$, at the $(k+1)$th iteration. S_{g}^{k} is the global tangent stiffness matrix and f_{g}^{k} is the unbalanced global force vector.

Having evaluated $\Delta\phi_{\mathrm{g}}^{k+1}$, the previous iteration values ϕ_{g}^{k} are updated using

$$\phi_{\mathrm{g}}^{k+1} = \phi_{\mathrm{g}}^{k} + \Delta\phi_{\mathrm{g}}^{k+1}. \tag{16}$$

Iterations are performed at each time step, in conjunction with moving the run-up tip and rezoning the mesh, to successively reduce the value of the components of f_{g}^{k}. Convergence is assessed by comparing the sum of the squares of the global unbalanced forces with the value of a preset tolerance. At the completion of iterations, f_{g}^{k+1} is the vector of nodal values of η and U which are the solution to (1) and (2) at $t = t^{i+1}$.

5.2. ALE method

This method uses a more sophisticated scheme for moving the nodes in which the co-ordinates vary continuously within a time step and previous time step values are not interpolated. This is achieved by treating the basis functions as functions of both space and time. Global time derivatives are therefore of the form

$$\frac{\partial\eta(x,t)}{\partial t} = \psi(x,t)\frac{\mathrm{d}\eta(t)}{\mathrm{d}t} + \frac{\partial\psi(x,t)}{\partial t}\eta(t), \tag{17}$$

in which the second term on the RHS arises owing to node movement. Equations (9) and (10) are therefore modified to

$$\int_{\Omega}\left(\omega\frac{\partial}{\partial t}(h+\eta) + \omega U\frac{\partial}{\partial x}(h+\eta) + \omega(h+\eta)\frac{\partial U}{\partial x} - \omega U_{\mathrm{e}}^{t}\frac{\partial}{\partial x}(h+\eta)\right)\mathrm{d}\Omega = 0, \tag{18}$$

$$\int_{\Omega}\left(\omega\frac{\partial U}{\partial t} + \omega U\frac{\partial U}{\partial x} + \omega g\frac{\partial\eta}{\partial x} - \omega U_{\mathrm{e}}^{t}\frac{\partial U}{\partial x}\right)\mathrm{d}\Omega = 0. \tag{19}$$

The last term on the LHS of (18) accounts for the change in size of the control volume used to derive (1) and is referred to here as the volume correction term. Similarly, the last term on the LHS of (19) accounts for the acceleration experienced by the control volume used to derive (2) and is referred to here as the convection correction term.

U_{e}^{t} is the 'velocity' of a particle moving along a line of constant ξ and is given by[10]

$$U_{\mathrm{e}}^{t}(\xi) = \frac{\mathrm{d}x(\xi,t)}{\mathrm{d}t} = \psi(\xi)\frac{\mathrm{d}x(t)}{\mathrm{d}t}. \tag{20}$$

Equation (2) is integrated numerically using the Euler method to give

$$U_{\mathrm{e}}^{t}(\xi) = \psi(\xi)\frac{x^{i+1} - x^{i}}{\Delta t}, \tag{21}$$

where x^{i} and x^{i+1} are nodal co-ordinates at $t = t^{i}$ and $t = t^{i+1}$ respectively.

A feature of the ALE method is that the global time derivatives are independent of the type of finite element approximation used in the spatial domain. In the model described here, the uncentred implicit formulation that was used with the DTI method is also used with the ALE method.

5.3. STE method

The STE method treats time as an extra spatial dimension. By using two-dimensional isoparametric elements, the type of co-ordinate transformation used in the spatial domain is extended to include the time domain. All quantities that are functions of time, including the nodal co-ordinates, are transformed in the time domain. Basis functions are functions of both space and time and the nodal values are constants for each element. Global derivatives are typically of the form

$$\frac{\partial\eta(x,t)}{\partial x}=\frac{\partial\psi(x,t)}{\partial x}\eta, \tag{22}$$

$$\frac{\partial\eta(x,t)}{\partial t}=\frac{\partial\psi(x,t)}{\partial t}\eta. \tag{23}$$

As for the DTI and ALE methods, the Newton–Raphson method is used to linearize the element equations. Only the components of f^k and s^k that are associated with local nodes at $t=t^{i+1}$ are evaluated, since the values of η and U at $t=t^i$ are known and are tested as initial conditions.

6. APPLICATION TO RUN-UP ON A SMOOTH SLOPE

The wave run-up problem shown in Figure 1 was used to demonstrate the deforming mesh along with the three methods of solving (1) and (2) (Table II). The initial length L equalled 8000 mm and the still water depth h equalled 520 mm. A sinusoidal water surface time series was specified at the seaward boundary, representing regular incident waves with wave height H equal to 40 mm and wave period T equal to 200 s.

The time step Δt was put equal to 1 s. The time-weighting parameter θ was put equal to 0·67 so as to give the highest order of accuracy (second order) achievable with the θ-method. A total time of 820 s was modelled.

Details of each mesh are listed in Table III. The first node of each mesh is located at the seaward boundary and the last node is located at the run-up tip. Mesh B is used with the STE method and consists of a row of two-dimensional elements defined in the space and time domains. Such a mesh is equivalent to placing two one-dimensional meshes Δt apart in the time domain. The first pair of node numbers listed in Table III therefore relates to the nodes at $t=t^i$ and the second relates to those at $t=t^{i+1}$. These co-ordinates are initial values and for subsequent time steps they depend on the node types, the distance moved by the run-up tip and, if relevant, the value of δ.

Computed time series for test NS11 are shown in Figure 4. As expected, there are oscillations within the first period of motion the transition from zero initial conditions to full periodic motion. The oscillations decay rapidly and the time series becomes both periodic and smooth. This implies that a simulation period equal to 4·1T is long enough to demonstrate that the solution is stable. The results are also in good agreement with the analytical solution of Sidén and Lynch,[11] with the RMS error equalling just 0·19 mm for $620 \leqslant t \leqslant 820$ s.

The distribution of nodes is shown in Figure 5. The right-hand edge of the plot is the locus of the run-up tip. The mesh expands and contracts during uprush and downrush respectively, with the elements deforming to account for the lateral movement of the run-up tip.

Table II. Test parameters

Test	Mesh	δ (*mm*)	Solution method
NS11	A	4	ALE
NS14	A	0	ALE
NS15	A	4	DTI
NS16	B	4	STE

Table III. Mesh details

Node no. (mesh A)	Node no. (mesh B)	Initial co-ord. (mm)	Node type
1	1, 2	0	E1
2	3, 4	200	E2
3–39	5, 6–77, 78	Linearly interpolated	E2
40	79, 80	7800	E2
41	81, 82	8000	E4

Test NS14 was performed to demonstrate the effect of the value of δ on node movement. Time series of the length of element 20, which is located next to the run-up tip, are compared in Figure 6 with those computed in test NS11.

Figure 6 shows that element 20 generally expands and contracts during uprush and downrush respectively. Because the node at the run-up tip is a type E4 node, its motion is not restricted by the value of δ and so it moves at all time steps. Element 20 is not part of a zone containing type E3 nodes and so the length of element 20 is constantly changing during both simulations. However, for test NS11 the other nodes directly attached to element 20 are moved only if the values of δ is exceeded (Figure 2).

Element 20 expands during uprush and then suddenly contracts as the node at the seward end of the element (node 39) is moved. The opposite occurs during downrush and these both cause the sporadic variations in element length for test NS11 near $t = 650$ and 750 s. The effect of using a non-zero value of δ is to restrain an internal node from being moved for one or more time steps and then to suddenly release it.

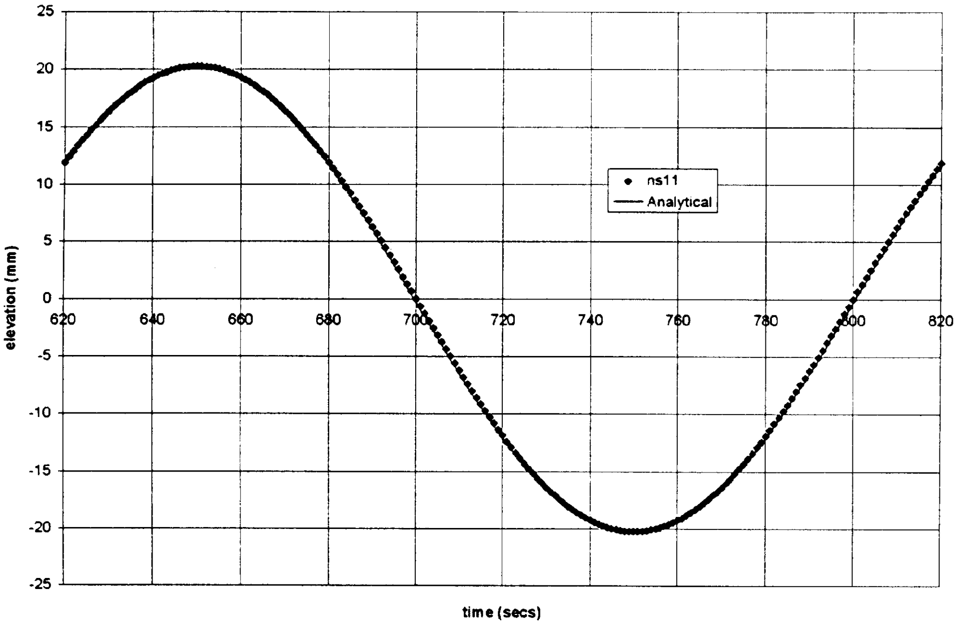

Figure 4. Time series of elevation of run-up tip (test NS11)

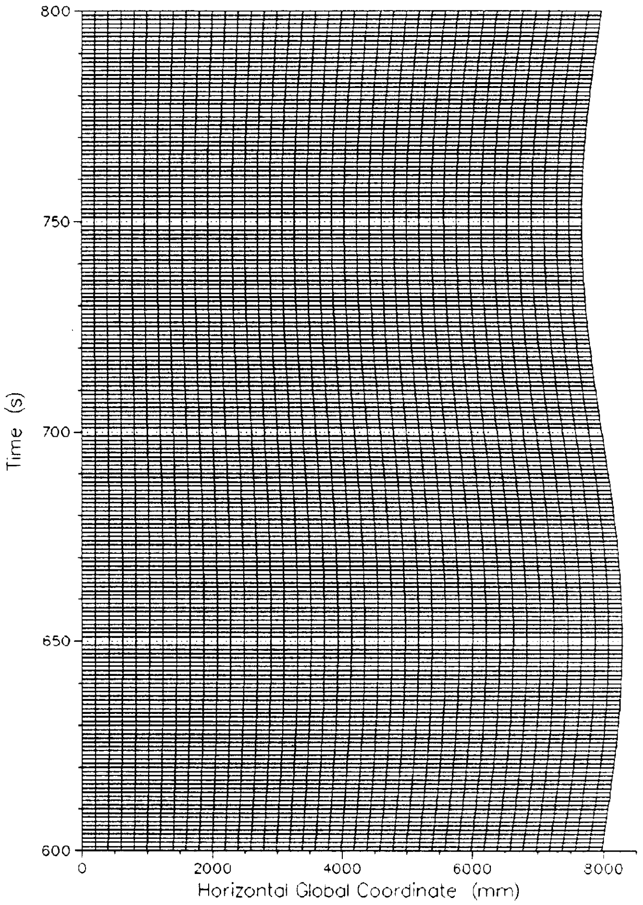

Figure 5. Space–time plot of mesh deformation (test NS11)

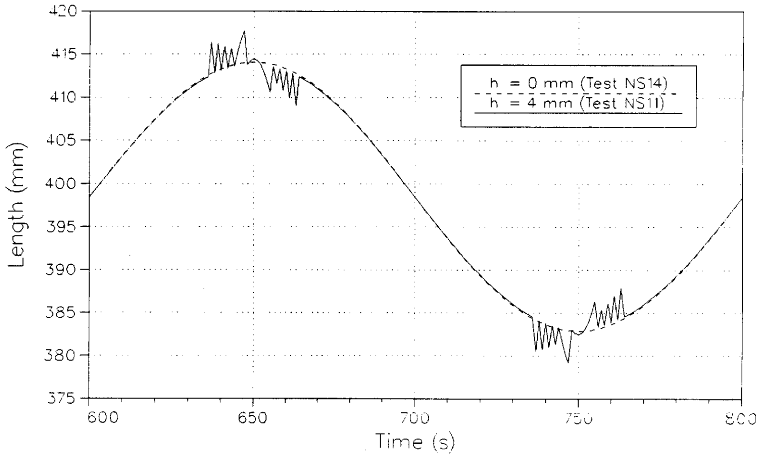

Figure 6. Time series of length of element 20

A detailed study of the number of recalculations required for each element of the mesh used in tests NS11 and NS14 was made for one wave period. The results show that placing a tolerance on node position through the specification of a non-zero value for δ reduces computational time without compromising the accuracy of the solution.

Tests NS15 and NS16 were performed using the DT1 and STE methods respectively (Table II). The results are in good agreement with the analytical solution of Sidén and Lynch,[11] with the RMS error for the elevation of the run-up tip equalling just 0·19 and 0·26 mm respectively for $620 \leqslant t \leqslant 820$ s.

To assess the effect of element deformation on the manner in which (1) and (2) are solved, time series for individual terms of (9) and (18) are computed for selected nodes. This simply involves substituting values of η and U into the unbalanced global force vector at the end of each time step. Results for test NS11 are shown in Figure 7.

The ALE method accounts for element deformation through the volume correction term. Because node 39 moves during the period $600 < t < 635$ s, the volume correction term is non-zero and the time series is smooth. This node is also moved to locations where the total water depth $h + \eta$ is similar to that at the previous time step. The term $\mathrm{d}(h + \eta)/\mathrm{d}t$ is therefore small relative to the volume correction term. Although h decreases with time as the node moves up the slope, the value of η increases at a slightly faster rate. The net effect is that $\mathrm{d}(h + \eta)/\mathrm{d}t$ is positive during this period. It is worth noting that the magnitude of the volume correction term is significant, implying that mesh deformation has a strong influence on volume (mass) continuity.

Sporadic motion of node 39 occurs during the period $635 < t < 665$ s and this causes the spiky appearance of the time series for both $\mathrm{d}(h + \eta)/\mathrm{d}t$ and the volume correction term. When node 39 is stationary, the magnitude of the former term increases and the value of the volume correction term reduces to zero.

To satisfy the continuity requirement of (1), a corresponding flux is required and this is represented in Figure 7 by the time series of the volume flux term.

Computed time series for test NS16 are shown in Figure 8. This test used the STE method and so element deformation is accounted for through the isoparametric co-ordinate transformations. This principal; difference between the STE and ALE methods can be seen by comparing Figures 7 and 8. With the ALE method a modified time derivative is calculated *within* the element equations, whereas

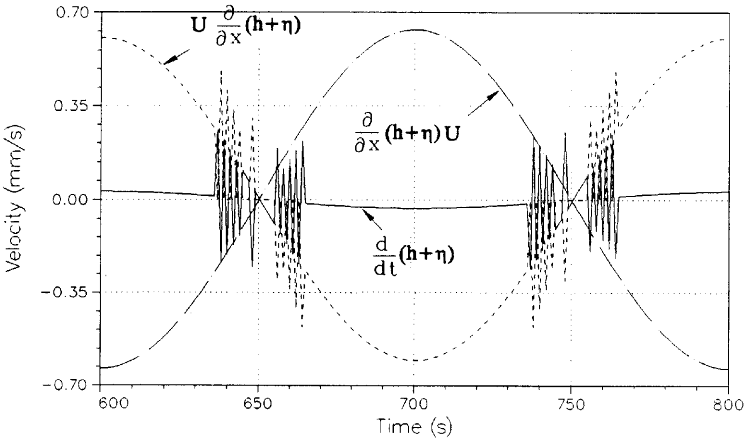

Figure 7. Time series of terms of continuity equation at node 39 (test NS11)

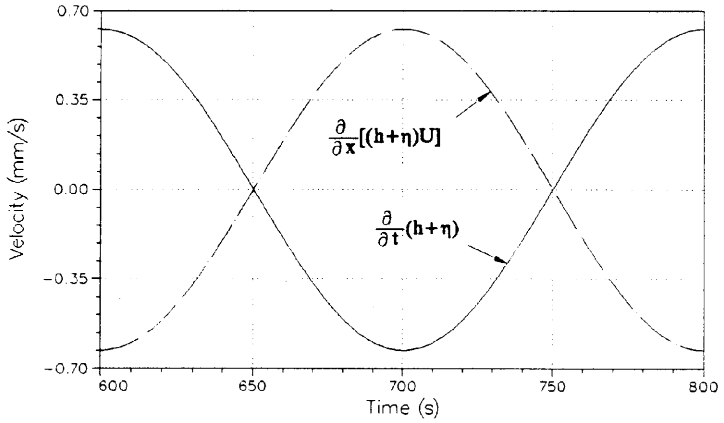

Figure 8. Time series of terms of continuity equation at node 39 (test NS16)

with the STE method the global time derivative is calculated *before* substitution into the element equations.[10]

7. DISCUSSION

The tests presented in this paper show that all three methods of solving (1) and (2) give good results for wave run-up on a smooth, mild slope. The DTI method uses a purely Eulerian form of (9) and (10) and therefore allows a fluid element formulation derived for a fixed mesh to be applied, unmodified to wave run-up. The ALE method gives the user flexibility in the choice of time integration, particularly when a finite difference method is desired. Because the STE method uses isoparametric elements within the time domain, a variable length time step can be introduced without having to modify the element equations. Although the STE method is potentially quite expensive computationally, this can be mitigated through the use of the mesh-rezoning scheme described here, which avoids non-essential recalculations through the judicious choice of node types.

The calculation of time histories of the terms of (1) as demonstrated here as practical application to the calculation of the forces associated with fluid–structure interaction. For example, many one-dimensional models account for bed roughness through the friction term in the momentum equation. The approach demonstrated here could be used to evaluate time series of this term and hence the fluid force acting on the seabed. It is also a useful way of checking that the volume (mass) and momentum conservation requirements of (1) and (2) have been satisfied.

In principle the model described here could be extended to two dimensions, which would have practical application to the run-up of obliquely incident waves. Motion of the shoreline would be represented by a two-dimensional kinematic boundary condition equation. This would be evaluated using local orthogonal velocities, avoiding the need to determine the local wave direction. The use of δ would be advantageous when dealing with the large number of nodes associated with multidimensional meshes, since it avoids non-essential node movement.

8. CONCLUSIONS

A novel approach to implementing a deforming finite element mesh has been demonstrated. One advantage of using a constant number of nodes and elements is that the global element matrix does not need to be restructured at each time step, making the solution method more efficient than earlier deforming mesh models. The use of a node position tolerance avoids node movement that would increase computational time without significantly improving the distribution of nodes. Rezone codes associated with each node type ensure that global derivatives are recalculated only when necessary.

The DTI, ALE and STE methods all give good results for wave run-up on a mild, plane slope. The calculation of time histories of the terms of the governing equations is a useful way of demonstrating how the ALE and STE methods account for mesh deformation. It also has practical application to the calculation of the forces associated with fluid–structure interaction and is a useful way of checking that the volume (mass) and momentum conservation requirements are satisfied.

The model described here could be extended to two-dimensions, which would have practical application to the run-up of obliquely incident waves.

APPENDIX: NOMENCLATURE

f_g^k	global vector of unbalanced forces
g	gravitational acceleration
h	water depth
H	wave height
I	time set counter
S	element stiffness matrix
t	time
U	depth-averaged horizontal velocity
U_e	element velocity
U_R	horizontal velocity at run-up tip
x	horizontal global co-ordinate
y	vertical global co-ordinate

Greek letters

δ	node position tolerance
η	water surface elevation
η^*	local vector of interpolated water surface elevation from previous time step
θ	time-weighting parameter
ρ	fluid density
ϕ	local vector of nodal values
ω	weighting function
Ω	domain of integration

REFERENCES

1. N. Kobayashi, A. K. Otta and I. Roy, ‘Wave reflection and run-up on rough slopes’, *J. Waterway, Port, Coastal, Ocean Eng., ASCE*, **113**, 282–298 (1987).
2. N. W. H. Allsop, J. V. Smallman and R. V. Stephens, ‘Development and application of a mathematical model of wave action on steep slopes’, in B. L. Edge (ed.), *Proc. 21st Int. Conf. on Coastal Engineering*, ASCE, Malaga, Spain, 1989, pp. 281–291.
3. J. W. van der Meer and M. K. Breteler, ‘Measurement and computation of wave induced velociteis on a smooth slope’, in B. L. Edge (ed.), *22nd Int. Conf. on Coastal Engineering*, ASCE, 1991, pp. 191–204.
4. T. C. Gopalakrishnan and C. C. Tung, ‘Numerical analysis of a moving boundary problem in coastal hydrodynamics’, *Int. j. numer. methods fluids*, **3**, 179–200 (1983).
5. J. G. Blom, J. M. Sanz-Serna and J. G. Verwer, ‘On simple moving grid methods for one-dimensional evolutionary partial differential equations’, *J. Comput. Phys.*, **74**, 191–213 (1988).

6. D. R. Lynch and W. G. Gray, 'Finite element simulation of flow in deforming regions', *J. Comput. Phys.*, **36**, 135–153 (1980).
7. R. Bonnerot and P. Jamet, 'A second order finite element method for the one-dimensional Stefan problem', *Int. j. numer. methods eng.*, **8**, 811–820 (1974).
8. G. N. Palmer, 'Numerical modelling of wave runup on breakwaters', *Thesis*, University of Auckland, 1994.
9. K. H. Huebner and E. A. Thornton, *The Finite Element Method for Engineers*, 2nd edn, Wiley, New York, 1982.
10. D. R. Lynch, 'Unified approach to simulation on deforming elements with application to phase change problems', *J. Comput. Phys.*, **47**, 387–411 (1982).
11. G. L. D. Siden and D. R. Lynch, 'Wave equation hydrodynamics on deforming elements', *Int. j. numer. methods fluids*, **8**, 1071–1093 (1988).