Calculation of Plunging Breakers with A Fully-Implicit Adaptive-Grid Method

Biing-Horng Liou* **Luigi Martinelli †** **Timothy J. Baker**

Department of Mechanical and Aerospace Engineering
Princeton University
Princeton N.J. 08544, U.S.A.

Antony Jameson ‡

Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, U.S.A.

1 Abstract

A new algorithm for simulating the plunging breaker is described. The approach is a fully-implicit multigrid scheme which uses the generalized artificial compressibility approach to couple the incompressible Euler equation and continuity equation in a hyperbolic manner. Several techniques for convergence acceleration are also implemented. The kinematic boundary is formulated by the Eulerian description which can easily deal with the open boundary. A grid adaptation strategy is used to capture the wave motions as well as the wavemaker's oscillations. Comparisons are made between the numerical solutions and the tank measurements for the plunging breaker generated by a piston-type wavemaker. Excellent agreement in the wave profile up to and including overturning validates the new method.

2 Introduction

Wave breaking is a very important hydrodynamic phenomenon. The breaking process introduces air bubbles, sprays, and noise and is responsible for the generation of water surface droplets. Breaking waves threaten the offshore structures as well as erode the coast with their abundant kinetic energy. For decades, calculations for such problems have been a challenging objective of many research efforts. Successful approaches, however, must be able to approximate the kinematic free surface boundary condition and track the free surface movement. There are two approaches available for tracking the interface, the front tracking technique and the front capturing technique.

The first approach is the front tracking technique which follows the interface directly. Two kinds of formulation, Lagrangian and Eulerian, are possible. The former seems to be the natural choice for describing the free surface location. However, it may develop a strong saw-tooth instability which was first reported by Longuet-Higgins and Cokelet [26] and, later, by several other investigators [18, 10, 41] unless a proper smoothing procedure or regridding technique is implemented. Moreover, when using a Lagrangian formulation to simulate open domain problems, new points have to be inserted upstream while the points expelled

*currently Postdoctoral Research Associate, Center for Simulation of Advanced Rockets, University of Illinois at Urbana-Champaign

†Professor of Mechanical and Aerospace Engineering, Member of AIAA

‡Thomas V. Jones Professor of Engineering, AIAA Fellow

out of the numerical boundary have to be removed. The alternative viewpoint, Eulerian, does not suffer from the same instability and can easily treat open domain problems. However, traditional Eulerian methods (i.e. describe wave height based on a fixed x-position or on a reference point movin g in the x-direction) become invalid when the waves start overturning. Therefore, a generalized Eulerian view point must be adopted to capture the breaking waves. One of the difficulties of the front tracking technique is that the numerical grid must be adapted (by moving the points, changing the connections between points, or altering the number of points). Nevertheless, the front tracking technique seems to be more accurate.

To implement this technique, suitable mesh movement and/or mesh-adaptation algorithms must be applied. Pure mesh movement can be achieved by a suitable mapping function or an artificial spring mechanism. For convenience, this is named the spring method to distinguish it from the particular adaptation method which means any adaptive mesh including reconnection, deletion, or addition of points. Details about mesh-adaptation strategies can be found in [4] and section 8.

The second approach is the front capturing technique which, instead of following the interface, constructs the free surface based on information such as the regions occupied by the fluids or updated scalar fields. The MAC method (Welch *et al.*, 1965 [43]), the VOF method (Hirt and Nichols, 1981 [16]) and the level set formulation (Osher,1988 [38]) are of this kind. This approach uses a fixed grid and thus is easier to implement. However, the front capturing technique is less accurate because it introduces some numerical viscosity near the interface.

We present a new scheme as follows. The front tracking approach is chosen owing to its accuracy. A new generalized Eulerian viewpoint is formulated to follow the free surface before plunging occurs. A fully-implicit scheme which generalizes the very efficient dual time-stepping scheme originally proposed by Jameson [20] for compressible flow is implemented to relax the time step limit. In the present algorithm, fast convergence to a steady state of the inner-iteration is achieved by making use of the multigrid technique developed by Hino, Martinelli and Jameson [17] for steady free-surface calculations on triangular grids. Details on the space discretization and a comprehensive study of the artificial compressibility method for unstructured grids is given in [11]. The A-stable discretization in time allows the stability constraint on the physical time step to be relaxed, while standard convergence acceleration techniques such as local pseudo-time stepping and residual averaging are applied to the pseudo-transient iteration. Also, to alleviate the stiffness stemming from the unsteady source terms in the residuals, a point-implicit five-stage Runge-Kutta scheme is constructed, following the guidelines given in [33]. Finally, the range of the characteristic wave speeds associated with the hyperbolic pseudo-transient problem is optimized for better convergence by employing a suitable form of local preconditioning [39, 6].

Two grid-adaptation techniques are also implemented. For plunging breakers, pure movement of the grid points seems inadequate. A procedure containing local remeshing, deletion of points and mesh enrichment must be used. In the literature, an adaptive-mesh technique was used by Löhner [28, 27] to follow the moving bodies in the flow. His enrichment procedure requires splitting triangles into two by putting points on their longest edges which might result in a locally high "aspect ratio" and have the tendency to have poor solution accuracy as reported in [24]. The adaptive scheme we used is based on Delaunay triangulation which maximizes the smallest angles of its triangles and usually results in better quality meshes[2, 3].

3 Governing Equations

Consider a general two-dimensional homogeneous incompressible inviscid free surface flow problem. The Cartesian coordinate system is chosen as the x axis being the still air-water interface while the y axis parallel to the gravitational force. Let the reference length, velocity and density be L_0, U_0 and ρ_0. The dimensionless Cartesian velocity components, dimensionless pressure and the dimensionless density are denoted by u, v, $\hat{p}$ and $\rho(=1)$ respectively. Let $p = \hat{p} + \frac{y}{F^2}$ be the pressure minus the hydrostatic part, where the Froude number $F = \frac{U_0}{\sqrt{gh}}$, h is height of the free surface, and g is the gravitational acceleration. The dimensionless governing equations which consist of the continuity equation and the time-dependent momentum equation are

$$\frac{d}{dt}\iiint_{V(t)} \rho dV + \iint_{S(t)} \rho(\mathbf{u}_r \cdot \mathbf{n}) dS = 0 \qquad (1)$$

and

$$\frac{d}{dt}\iiint_{V(t)} \rho \mathbf{u} dV + \iint_{S(t)} \rho \mathbf{u}(\mathbf{u}_r \cdot \mathbf{n}) dS = -\iint_{S(t)} p\mathbf{n} dS \qquad (2)$$

where $\mathbf{u}$ is the velocity measured with respect to an inertial reference frame , $\mathbf{u}_r = \mathbf{u} - \mathbf{u}_b$ is the velocity of the fluid relative to the control surface with velocity $\mathbf{u}_b$, and $\mathbf{n}$ is the unit normal of the control surface. Note that the the body force is included in the pressure term

in equation (2). A more compact form of equations (1) and (2) is

$$\frac{d}{dt}\int\int_{V(t)} \mathrm{t}dxdy + \int_{S(t)} (\mathrm{f}dy - \mathrm{g}dx) = 0, \qquad (3)$$

where $\mathbf{t} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}$, $\mathbf{f} = \begin{pmatrix} \rho u_r \\ \rho u u_r + p \\ \rho v u_r \end{pmatrix}$, $\mathbf{g} = \begin{pmatrix} \rho v_r \\ \rho u v_r \\ \rho v v_r + p \end{pmatrix}$, and $\mathbf{w} = \begin{pmatrix} p \\ \rho u \\ \rho v \end{pmatrix}$.

Equation (3) gives three equations in three unknowns $\mathbf{w}$.

3.1 Initial and Boundary Conditions

3.1.1 Solid Surface

The free-slip condition on the solid bodies is implemented by

$$\mathbf{u}_r \cdot \mathbf{n} = 0 \qquad (4)$$

which imposes no normal flow through solid boundaries.

3.1.2 Free Surface

The free surface condition consists of a dynamic and a kinematic condition. The former, which is known as the DFSBC (Dynamic Free Surface Boundary Condition), states the continuity of normal stress on the air-liquid surface. For the inviscid case without considering surface tension, this is expressed as

$$\hat{p} = p_0, \quad \text{at} \quad y = h(x,t)$$

or

$$p = p_0 + \frac{h}{F^2}, \quad \text{at} \quad y = h(x,t) \qquad (5)$$

where p_0 is the atmospheric pressure (assumed to be constant). The latter condition or KFSBC (the Kinematic Free Surface Boundary Condition) describes the free surface on a material surface and can be formulated either by the Lagrangian or the Eulerian viewpoints.

The Lagrangian point of view is a natural extension of particle mechanics and describes the particle locations at the free surface by

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}. \qquad (6)$$

Alternatively, the Eulerian formulation, which is widely used in fluid mechanics, describes a fixed point in space with respect to a reference frame. The free surface location (h_x, h_y) is written as

$$\frac{\partial h_y}{\partial t} + u_r \frac{\partial h_y}{\partial x} - v = 0, \quad \text{at} \quad y = h_y \qquad (7)$$

where $u_r = u - u_b$ is the x component of relative velocity, provided that $u_b = \frac{\partial h_x}{\partial t}$ is given. This particular eq uation is unable to follow overturning waves because h_y will become multivalued for given h_x. Nevertheless, equation (7) is widely used in steady problems.

There is a way to describe overturning waves using the Eulerian formulation. Suppose a general coordinate system, $O-\xi-\eta$ is chosen with base $(\mathbf{e}_\xi, \mathbf{e}_\eta)$. The free surface location $h_\xi \mathbf{e}_\xi + h_\eta \mathbf{e}_\eta$ can be obtained by solving

$$\frac{\partial h_\eta}{\partial t} + u_{\xi r}\frac{\partial h_\eta}{\partial \xi} - u_\eta = 0, \quad \text{at} \quad \eta = h_\eta, \qquad (8)$$

where $u_{\xi r} = u_\xi - u_{\xi b}$ is the ξ component of relative velocity, provided that $u_{\xi b} = \frac{\partial h_\xi}{\partial t}$ is given. There is no need to use a single $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ for all points and/or at all time. At a fixed time, each $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ is chosen at each point to guarantee single-valueness of h_η.

3.1.3 Initial Condition

Initially, there is no flux, the free surface is undisturbed,

$$u = 0, v = 0, p = p_0, h = 0 \qquad (9)$$

4 Discretization

The space domain is subdivided into a number of cells (triangles). The governing equations are solved for each control volume which consists of cells with a common point. Figure 1 shows the control volumes for interior and boundary points. The hollow circles represent the locations of residual collection, the solid circle circles are the contributing nodes, and the grey areas are the control volumes. For boundary control volumes, points where residuals are collected are also contributing nodes, and these are referred to as *one-sided* control volumes.

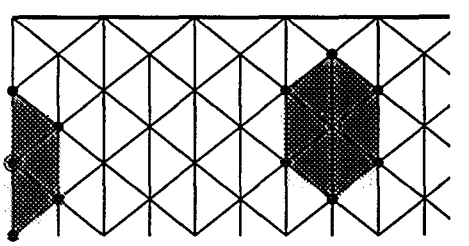


Figure 1: Control volumes for boundary and interior points

A vertex-centered formulation of (3) yields the semi-discrete equations

$$\frac{d}{dt}[\mathrm{t}_i V_i] + \mathbf{R}_i(\mathbf{w}_i) = 0, \qquad (10)$$

where the residual

$$\mathbf{R}(\mathbf{w_{ij}}) = \mathbf{E}(\mathbf{w_{ij}}) + \mathbf{D}(\mathbf{w_{ij}})$$

is obtained by approximating convective fluxes with central differencing in space plus a third order artificial dissipation term to prevent an odd-even decoupling. The dissipation term $\mathbf{D}(\mathbf{w_{ij}})$ is obtained (after adding the generalized artificial compressibility terms) by

$$\mathbf{D}_i(\mathbf{w}_i) = \sum_j \epsilon^4_{ij}(\nabla^2_u \mathbf{w}_j - \nabla^2_u \mathbf{w}_i)$$

where the coefficient ϵ^4_{ij} is

$$\epsilon^4_{ij} = \mu_4 \, \rho_{ij},$$

μ_4 is an arbitrary parameter, the undivided Laplacian is

$$\nabla^2_u \mathbf{w}_i = \sum_j (\mathbf{w}_j - \mathbf{w}_i)$$

and ρ_{ij}, the spectral radius of the edge with end points i and j is taken as

$$\rho_{ij} = \min(\rho_i, \rho_j).$$

The spectral radius ρ_i can be shown as of the form

$$\rho_i = \max(|U_i - q_i|, |U_i - \frac{1}{2}q_i| + a_i),$$

where $U_i = u_i \Delta y_i - v_i \Delta x_i$, $q_i = u_{bi}\Delta y_i - v_{bi}\Delta x_i$, and $a_i = \sqrt{(U_i + \frac{q_i}{2})^2 + \Gamma^2(\Delta x_i^2 + \Delta y_i^2)}$.

It is easily seen that with the artificial dissipation terms constructed, the total space discretization

$$\mathbf{R}(\mathbf{w}_i) = \{\sum_j (\mathbf{f}_j \Delta y_j - \mathbf{g}_j \Delta x_j)\} + \mathbf{D}_i(\mathbf{w}_i)$$

is still in conservative form provided that the spectral radii are constant.

A backward difference discretization in time of the form

$$\frac{d}{dt} = \frac{1}{\Delta t} \sum_{q=1}^{k} \frac{1}{q!} \left[\Delta^-\right]^q, \quad \Delta^- = (\cdot)^{n+1} - (\cdot)^n \qquad (11)$$

is used. In particular, dropping the subscripts i, j for clarity, for a second order discretization in time one obtains

$$\mathbf{R}^*(\mathbf{w}) = \frac{1}{\Delta t}\left(q_1[\mathbf{t}\,V] + q_2[\mathbf{t}^n V^n] + q_3[\mathbf{t}^{n-1}V^{n-1}]\right) + \mathbf{R}(\mathbf{w}),$$

where $\mathbf{q} = (2, -3/2, 1/2)$. Application of the generalized artificial compressibility approach results in the system of coupled O.D.E.s to be solved to convergence on every time step

$$\frac{d(\mathbf{w}V)}{dt^*} + \mathbf{Pr} \cdot \mathbf{R}^*(\mathbf{w}) = \mathbf{0}, \qquad (12)$$

where the diagonal preconditioning matrix is $\mathbf{Pr} = \text{diag}\,[\Gamma^2, 1, 1]$ and $\Gamma^2 = \max(0.3, 3(u^2 + v^2))$.

A point-implicit k-stage Runge-Kutta method for the pseudo-time iteration can be cast as

$$\mathbf{w}^{(0)} = \mathbf{w}^n$$

$$\cdots$$

$$\mathbf{w}^{(i)}(1 + \alpha_i a_1 \frac{\Delta t^*}{\Delta t}\mathbf{I}^m) = \mathbf{w}^{(i-1)} - \alpha_i \Delta t^* [\frac{1}{V}\mathbf{R}^{*(i-1)} - \frac{1}{\Delta t} a_1 \mathbf{w}^{(i-1)} \cdot \mathbf{I}^m]$$

$$\cdots$$

$$\mathbf{w}^{n+1} = \mathbf{w}^{(k)}$$

where $\mathbf{I}^m = diag\{0, 1, 1\}$ and $\mathbf{w}^n$ is the value of $\mathbf{w}$ after n pseudotime steps, is applied to drive equation (12) to steady state in pseudotime t^*. Once a steady state is reach, equation (12) is satisfied and one step in real time has advanced. For most of this work, a very efficient five-stage Runge-Kutta is used. Table 1 lists the coefficients.

	ν =1	2	3	4	5
α_ν	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{3}{8}$	$\frac{1}{2}$	1

Table 1: 5-stage Runge-Kutta scheme

4.1 Kinematic free surface boundary condition

The kinematic free surface boundary condition (KFSBC) can be formulated either by a Lagrangian or by an Eulerian viewpoint. For the former case, the location of the free surface is obtained by computing

$$\frac{d\mathbf{x}_i}{dt} = \overline{\mathbf{u}}_i \qquad (13)$$

where $\overline{\mathbf{u}}_i = \frac{1}{16}(-\mathbf{u}_{i-2} + 4\mathbf{u}_{i-1} + 10\mathbf{u}_i + 4\mathbf{u}_{i+1} - \mathbf{u}_{i+2})$ is the smoothed velocity vector as suggested by [26].

For the Eulerian approach, the convective term of the η component of the KFSBC is approximated by a central difference in space plus a third order artificial dissipation term. In order to keep the ratio between

two adjacent edges' lengths constant, a spring method is formulated on the ξ component,

$$\frac{\partial h_{\xi i}}{\partial t^*} = k_1 \Delta s_{i+\frac{1}{2}} - k_2 \Delta s_{i-\frac{1}{2}} \tag{14}$$

where $\Delta s_{i+\frac{1}{2}} = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$ is the length of the edge with end points i and $i+1$ and $\frac{k_1}{k_2}$ is a constant.

No matter what viewpoint is taken, the time-marching equation for the KFSBC must be solved together with the bulk flow solver. It is discretized in space and then solved by implicit time stepping. A pseudo-time derivative term is then added to implement the dual time stepping method described earlier.

5 Convergence Acceleration

Several efficient techniques are employed to accelerate convergence at each time step. The most important one is the multigrid scheme, which also uses separately generated meshes. The details of the multigrid scheme can be found in [32, 21]. Another is called the local time stepping technique which allows each control volume to be advanced in pseudo-time by its own maximum local pseudo-time step. Residual averaging is also an effective method to increase the pseudo-time step by collecting information from residuals at neighboring points [21]. These techniques are performed on the subiterations and do not affect time accuracy.

5.1 Unstructured Multigrid

The multigrid method was first proposed by Fedorenko in 1964 [15] for elliptic equations, and some promising results were obtained later. The method we used here is based on Jameson's 1985 method[21].

The multigrid procedure begins at the finest grid. Let the initial value $\mathbf{w}_1^{(0)}$ be the flow value at pseudo-time level n. The equation

$$\frac{d\mathbf{w}_1 V}{dt^*} + \mathbf{R}^*(\mathbf{w}_1) = 0$$

is solved for $n+1$ by a point-implicit Runge-Kutta method. The solution is transferred to a coarser grid by a transfer operator P_h^{h+1} where h denotes the grids. The initial condition on the grid $h+1$ is

$$\mathbf{w}_{h+1}^{(0)} = P_h^{h+1} \mathbf{w}_h.$$

Then on grid $h+1$, the updated solution is obtained by forwarding

$$\frac{d\mathbf{w}_{h+1} V_{h+1}}{dt^*} + \mathbf{R}^*(\mathbf{w}_{h+1}) + \mathbf{G}_{h+1}(\mathbf{w}_h, \mathbf{w}_{h+1}^{(0)}) = 0 \tag{15}$$

where

$$\mathbf{G}_{h+1}(\mathbf{w}_h, \mathbf{w}_{h+1}^{(0)}) = Q_h^{h+1} \mathbf{R}^*(\mathbf{w}_h) - \mathbf{R}^*(\mathbf{w}_{h+1}^{(0)}).$$

For simplicity, we split the modified residual $\mathbf{R}^*(\mathbf{w})$ according to

$$\mathbf{R}^*(\mathbf{w}) = \mathbf{L}(\mathbf{w}) + \mathbf{S}(\mathbf{t}^n; q),$$

where

$$\mathbf{L}(\mathbf{w}) = a_0 \mathbf{t}^{n+1} V^{n+1} + \mathbf{R}(\mathbf{w})$$

and

$$\mathbf{S}(\mathbf{t}^n; q) = \frac{1}{\Delta t} \sum_{k=1}^{q} a_k \mathbf{t}^{n+1-k} V^{n+1-k}$$

is the fixed source term from the previous time level. q is the order of accuracy of the physical time stepping. Since this source term is identical for $\mathbf{R}^*(\mathbf{w}_{h+1})$ and $\mathbf{R}^*(\mathbf{w}_{h+1}^{(0)})$, equation (15) can be written as

$$\frac{d\mathbf{w}_{h+1} V_{h+1}}{dt^*} + \mathbf{L}(\mathbf{w}_{h+1}) + Q_h^{h+1} \mathbf{R}^*(\mathbf{w}_h) - \mathbf{L}^*(\mathbf{w}_{h+1}^{(0)}) = 0.$$

Consequently the coarser grids at previous time steps are of no influence. Finally the corrections to the solution on the finer grid are obtained by

$$\mathbf{w}_h^{new} = \mathbf{w}_h^{(0)} + I_{h+1}^h (\mathbf{w}_{h+1} - \mathbf{w}_{h+1}^{(0)}).$$

Details of the transfer operators P_h^{h+1}, I_{h+1}^h, and Q_h^{h+1}, can be found in [23, 17].

6 Unstructured Grid Techniques

The four most popular unstructured mesh generation techniques are the advancing front, Delaunay triangulation, edge-face swapping and the octree-based techniques. The main feature of the advancing front method is that elements (i.e. triangles or tetrahedra) and points are generated at the same time. However, it poses some difficulties when the new point is very close to an existing point of the front, when the new triangle intersects with the nearby edges of the front, or when two edges of very different length come close to each other. The Delaunay triangulation method is supported in two dimensions by solid theoretical results and can easily handle a new point to be inserted at any time. The edge-face swapping technique needs an existing triangulation but can be utilized to convert a mesh to an ideal one. The octree-based technique uses repeated subdivisions of a given set of square elements.

6.1 Delaunay Triangulation

For a given set of N points $\{P_i\}$ in a region D, the Delaunay triangulation is the unique triangulation which has the property that no points lie inside the circumcircle of any triangle. To construct the Delaunay triangulation, let's consider the Voronoi region, which is the region closer to each point than any other. Mathematically, the Voronoi region is defined as $\bigcap_{i \neq j} H_{ij}$ where

$$H_{ij} = \{x \in R^3 | d(x, P_i) \leq d(x, P_j)\}$$

and $d(x, P_i)$ is the Euclidean distance. The union of the Voronoi regions is called Voronoi diagram or Dirichlet tessellation which is the geometric dual of Delaunay triangulation. The Delaunay triangulation is the triangulation formed by connecting points whose Voronoi regions have a common face.

The method to generate a mesh is based on an initial triangulation of the boundary points followed by the insertion of new points inside the domain according to a prescribed rule. The complete algorithm is described in the next section.

7 Algorithm to Construct the Delaunay Triangulation

The algorithm presented below follows the work of Baker [2] which is a sequentially construction method and is readily applied to any dimensions.

STEP 1 Define the boundary points and surfaces.

STEP 2 Initially triangulate the boundary points.

STEP 3 Choose interior points to be inserted according to the threshold value.

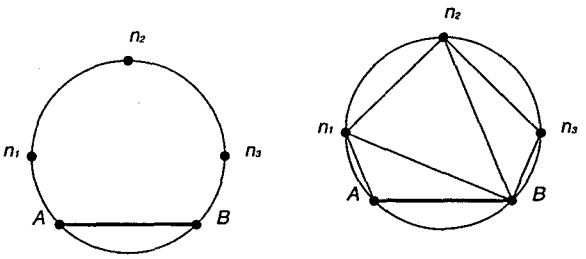
STEP 4 Introduce the points chosen in STEP 3 to the domain by the Bowyer-Watson Algorithm and using quadtree data structure to enhance this process.

STEP 5 Repeat STEP 3 and STEP 4 until no points are to be inserted.

STEP 6 Smooth the mesh.

7.1 Initial Triangulation

To start the mesh generation process, a given number of boundary points and its grid connections (edges, in planar case) are defined. (For the three-dimensional case, boundary grids can be constructed by using a two-dimensional triangular mesh generator.) It is convenient to define *eligible triangles* as the triangles whose edges do not intersect with the boundary edges and which lie internal to the domain (inside the external boundary and outside the internal boundaries). When a point can be connected to an edge to form a triangle which is eligible, this point is *eligible*. The initial triangulation implemented here is to assign a point on the boundary to each existing edge to form a triangle. For a given edge, the point is chosen based on the smallest circumradius of all the triangles formed by eligible points and the edge under consideration.



2a: five points are cyclic with base $\overline{AB}$

2b: reconstruct triangles based on the angle with $\overline{AB}$

Figure 2: Degenencies

It is possible, however, that degeneracies will occur. For a planar case, four or more points are cyclic, i.e. they are on a circle and no other points lie inside the circle. Degeneracies will make the triangulation non-unique. Even though uniqueness is not a concern for the generation of a computational grid, modifications must be made, otherwise the triangles could overlap. Suppose that $\overline{AB}$ is the edge under consideration and points A, B, n_1, $\cdots$, n_p are cyclic as in figure. n_i is presorted such that $\theta_1 > \theta_2 > \cdots > \theta_p$, where $\theta_i = \angle n_i AB$. The triangles formed are chosen as ΔABn_1, ΔBn_1n_2, $\cdots$, $\Delta Bn_{p-1}n_p$. Figure 2 shows an example with five points cyclic.

7.2 Mesh Refinement and Bowyer/Watson Algorithm

Once the initial triangulation is obtained, a mesh can be refined by introducing points inside the domain and reconstructing the triangulation to include the new points based on the Delaunay criteria. To achieve this, several algorithms have been developed. Among them, the Bowyer [7] and/or Watson [42] algorithm is straightforward and is easily extended to any number of dimensions. The procedure is as following. First all the triangles whose circumcircles contain a new point are removed. These removed triangles form a cavity in the domain. Second, the new point is connected with all the nodes on the boundary of the cavity to form new tri-

angles.One can see that the procedure meets the properties of the Delaunay triangulation. Therefore, the resulting mesh is Delaunay.

It should be noted that the algorithm can be implemented without difficulty for degenerate cases.

7.3 Point Selection

To complete the grid generation, points must be chosen and then inserted by the Bowyer-Watson algorithm. For many earlier applications, points were determined by some prescribed techniques somewhat like the selection of points on structured meshes. Recently, a different approach has been developed, which chooses points simultaneously with the construction of the grid. We use the method suggested by Rebay [40] which belongs to the latter category.

The new points are chosen as the circumcenters of all existing triangles if the triangles are acute, or as the midpoints of the longest edge if the circumcenters are outside their triangles. The points are sequentially inserted until some threshold value is larger than all circumradii. Typically the threshold value is location dependent in order to reflect the need for varying mesh size. For a boundary point P, the threshold value f_P is selected as the average length of two boundary edges with the common end point P. When a new point is introduced, its threshold value is taken as the area average of the threshold values at the vertices of the triangle which contains the new point.

7.4 Mesh Point Smoothing

The unstructured grid generated in this manner can be smoothed to decrease grid skewness. A popular way is to use a Laplacian filter which smoothes each point K times according to

$$\mathbf{x}_0^{k+1} = \mathbf{x}_0^k + \frac{\mu}{m} \sum_{i=1}^{m} \mathbf{x}_i^k - \mathbf{x}_0^k$$

where $\mathbf{x}_0^{k+1}$ is a new smoothed position of point $\mathbf{x}_0^k$, μ is the relaxation parameter, and m is the total number of neighbors with coordinates $\mathbf{x}_i$ surrounding $\mathbf{x}_0$.

After such a smoothing process, however, the mesh is not a Delaunay triangulation in general. Nevertheless, by implementing the adaptation method which will be described in the next chapter, the mesh can be restored to a Delaunay triangulation but with better grid quality.

8 Adaptive-Mesh Strategies

Adaptive-mesh strategies have been applied to problems either to follow movements of the boundaries or to improve the numerical solutions. Generally speaking, a strategy involves one or more of the following processes to modify the grids: by changing the number of points (h-method), by rearranging the distribution of points (r-method) and by altering the connectivity of points (m-method).

For the current study, the main issue is the inevitable deformation of the physical domain due to the motion of the free surface. Two grid-adaptation strategies are implemented and tested. One is the spring method (r-method) which uses a spring mechanism to deform the mesh nodes. This method ignores the remeshing procedure and therefore takes into account only mesh deformations. When the mesh distorts too much, the accuracy of the numerical solution may suffer. The other strategy is the adaptation method which inserts points inside the domain when better resolution is required and deletes existing points when these points lie outside the domain. Therefore, point-mesh rearrangement is taken care of and the mesh quality can be guaranteed.

For unsteady flow problems with large movement of boundaries, i.e. breaking waves, the spring method was simply abandoned in this work. The reason is that a point which initially lies outside any triangle might move inside a triangle and therefore invalidate the mesh. Furthermore, even if this does not happen, when the mesh distorts, the computational results will suffer [24]. Owing to such considerations, only the adaptation method is investigated here.

8.1 Adaptation Method (r,m,h-method)

This method contains the following steps.

Mesh movement (r-method)

Since the velocities of the boundary points are prescribed, the interior points are moved according to Laplace's equation

$$\nabla^2 \mathbf{u}_b = 0, \tag{16}$$

which guarantees the mesh velocities to be smoothly varying.

Addition of boundary points (h-method)

A new boundary point is added at the center of the boundary edge when the slope of this edge is less than -1. After the wave overturns, points are added near the

tip of the overturning wave until the spacing between two adjacent points are $\frac{1}{8}$ of the original spacing.

Local remeshing (m-method)

- identify the regions to be regenerated (failed the Delaunay test)
- remove all the points inside these regions
- re-introduce the points into the domain

Deletion of points (h-method)

- identify the points to be deleted according to the local threshold value
- remove the points and the edges connected to those points
- retriangulate the regions based on the algorithm of initial triangulation

Mesh enrichment (h-method)

- identify the points to be added according to 7.3
- insert the points using the Bowyer-Watson algorithm

9 Results

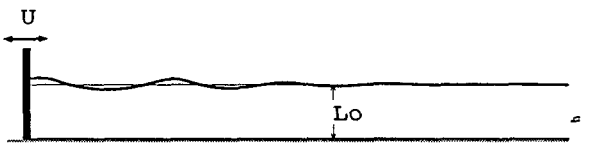


Figure 3: Characteristics of wavemaker problem

Consider a fluid in a finite rectangular tank with a vertical piston wavemaker at one end as shown in figure 3. The characteristic length, velocity, and density are set as the initial depth of the water H, $\sqrt{gH}$, and the fluid density. The nondimensional length of the tank is 20 in this study, and the origin of the axis is fixed at the initial point of the intersection between wavemaker and unperturbed free surface. The horizontal velocity of the piston wavemaker $U(t)$, as shown in figure 4, is prescribed as the following Fourier series

$$U(t) = \sum_{n=1}^{72} U_n \cos(\omega_n t - \theta_n) \qquad (17)$$

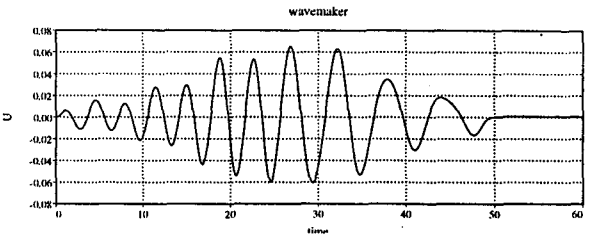


Figure 4: Motion of wavemaker

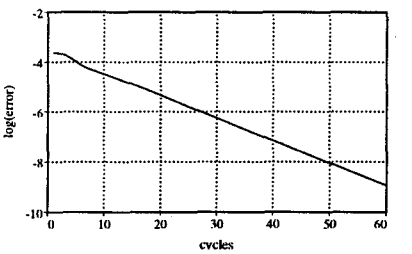


Figure 5: Convergence History of a Single Time Step

The amplitudes U_n, frequencies ω_n, and phase θ_n are tabulated in [10].

The convergence history plots shown in figure 5 indicate that with the point-implicit five-stage Runge-Kutta scheme, 15-30 multigrid cycles are sufficient to make the velocity field solenoidal within a tolerance of 10^{-6}.

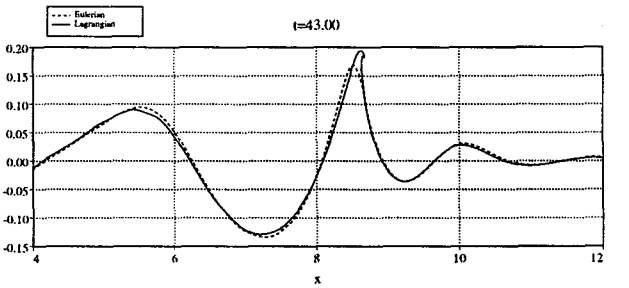
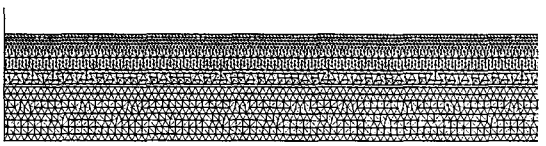


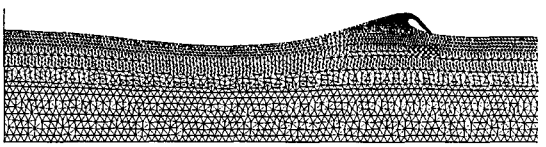
Figure 6: Part of the Wave Profiles Computed with Two Different Approaches at $t = 43.0$

Two formulations of the kinematic free surface boundary condition, Lagrangian and Eulerian, are considered. Fig 6 shows the wave profile at $t = 43$ for both formulations. There is a significant difference. The wave is overturning with the former formulation while it is not with the latter formulation, and is not in the experiments. The earlier overturning is predicted with the Lagrangian description no matter how fine the grid and no matter what kind of time-stepping

scheme is used. Earlier investigators, when using the Lagrangian approach, used a smoothing procedure to even out the velocities and wave height in order to damp the so-called saw-tooth instability and did not seem to have the same problems we observed. This is probably because the smoothing technique introduces dissipation terms in the calculation of the velocities and wave height just large enough to prevent this physically incorrect phenomena from happening. However, such a smoothing procedure cannot be easily applied to formulations other than the boundary integral method. This is because for a panel method smoothing the velocities at the free surface will affect the velocities inside the domain. On the other hand, for formulations with interior points, a suitable smoothing technique should smooth all the velocities in the domain and this is rather difficult (there is no indication of how to smooth them). Moreover, the calculation shown in reference [10] overpredicted the wave height at $t = 42$. So, the Lagrangian formulation of the free surface motion was simply abandoned in this work. The calculations done hereafter are based on the Eulerian description of the kinematic free surface boundary condition. One advantage of this formulation is that it can describe problems with an open free surface boundary without addition and deletion of free surface nodes.



7a: Initial Finest Grid (9850 points)



7b: Final Finest Grid, $t = 51.57$ (10699 points)

Figure 7: Part of Initial and Final Time Steps of Finest Grids, $8 \leq x \leq 13$

Four levels of multigrid are used for the computation. Figure 7 shows the initial and final ($t = 51.57$) finest grid. It indicates that our grid adaptation method together with the unsteady flow solver can handle the flow after the plunging occurs. Figure 8 shows the four levels of grid at time $t = 51.27$. The finest grid at this instant holds 10476 points. Each grid is generated and adapted without reference to the other grids. Furthermore, the meshes shown are Delaunay triangulations which possess the max-min angle property and thus can prevent inaccuracies due to the grid skewness

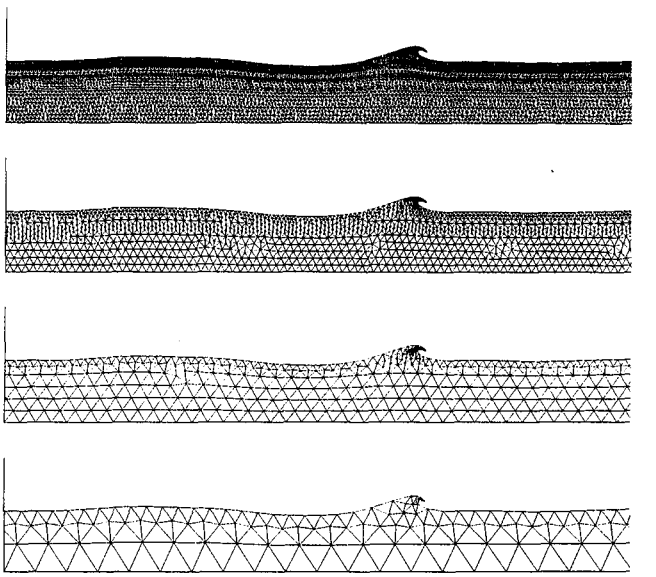
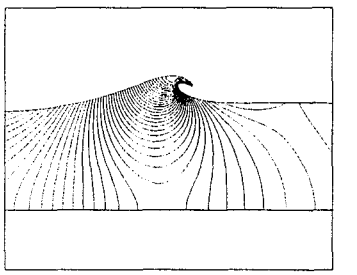


Figure 8: Four Levels of Grid at Time $t = 51.27$ and Located Between $x = 5$ and $x = 15$

9a:



9b:

Figure 9: Pressure (p) Distribution at Time $t = 51.27$

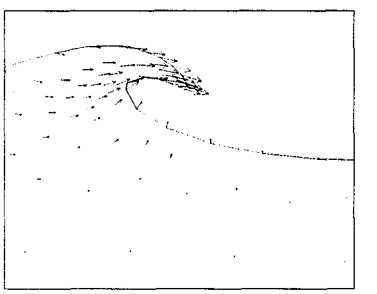
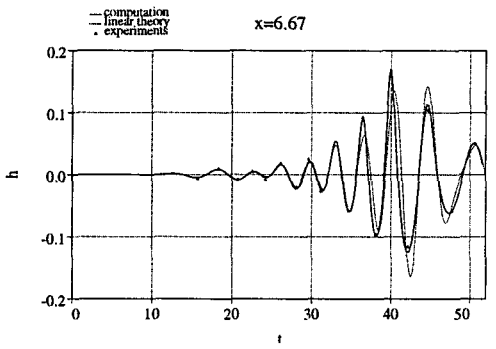


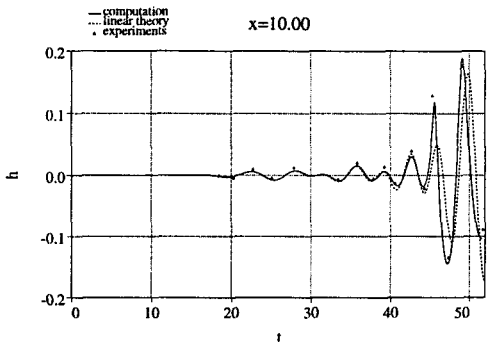
Figure 10: Velocity Vector Near the Breaker at Time $t = 51.27$

as reported in [24].

The pressure distribution at $t = 51.27$ is plotted in figure 9a and a close up view near the breaker is shown in figure 9b. The velocity vectors at time $t = 51.27$ shown in figure 10 indicate that the maximum speed of the plunging wave is located in the cusp, which is approximately twice the phase speed, with the central frequency of $\omega_c = 1.4$ based on the linear theory. The figure also indicates that the particles on the back of the breaker are moving away from it.



11a: $x = 6.67$



11b: $x = 10.00$

Figure 11: Wave Height Comparison of Nonlinear Computation, Linear Theory, and Wave-Probe Measurement at Location 6.67 and 10.00.

The free surface profiles at several different locations, i.e. $x = 6.67$ and 10.00, are compared with linear theory and measurements in figure 11. It shows that our unsteady solver is able to capture the phase as well as the amplitude of the wave motion. The linear theory can follow the phase of the waves, but the discrepancies for the amplitude based on the linear theory become larger as the wave group propagates downstream.

The numerical free surface heights near the plunging location at times $t = 51.11$, 51.24, 51.34, and 51.55 are plotted in figure 12 The measured surface heights at $x = 11.83$ at the same time instant are also shown.

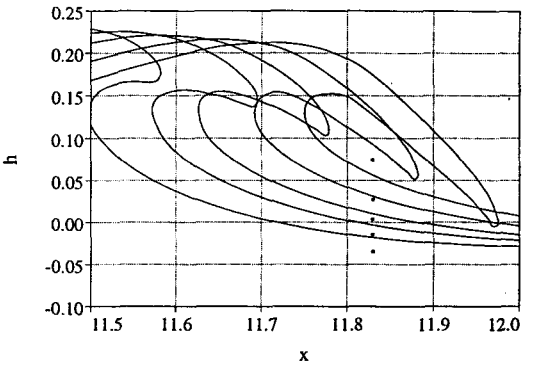


Figure 12: Instantaneous Free Surface Profile at Times $t = 51.11$, 51.24, 51.34, and 51.55 from Left to Right Respectively. Solid Rectangles denote the Measured Surface Height at $x = 11.83$ at the Same Time Instant from Bottom to Top Respectively.

Our numerical solutions predict wave heights in close agreement with the experiments. The maximum deviation occurs at $t = 51.55$, which, according to our calculation, is after the re-entry takes place. The computed plunging location and time differ from the experiments by about -0.16 and -0.30. Approximating the open far end by a solid wall in the simulation contributed to these errors. Dommermuth *et al.*[10] surmised that another factor could be the dissipation effect in their experiments which will delay the breaking process. On the bottom of the breaker, the numerical solution exhibits some small-scale disturbances. These disturbances might be physical because in that part in the overturning wave, a Rayleigh-Taylor instability is observed in the experiments.

Comparison with Boundary-Integral Methods

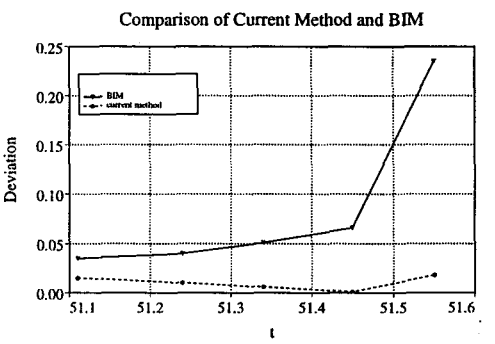


Figure 13: The difference between computed and measured free surface height at x=11.83

Prior to this work, boundary integral methods (BIMs) were considered to be the only successful method to simulate overturning waves. BIM methods have several nice properties. Because there are no internal grid points, BIM methods are thought to be the most efficient and accurate. Here we compare several results between our method and the BIM (for the same number of boundary points).

For the current test case, our computed results are compared with the boundary integral method by Dommermuth *et al.*[10]. The deviation of the computed from measured free surface elevations at x=11.83 are plotted in figure 13. It shows that our method predicts free surface locations closer to the experimental result than this boundary integral method. For t=51.55, the overturning wave predicted by the BIM has passed $x = 11.83$, and therefore the deviation goes up abruptly. For the present calculation, the upper fluid plunges into the lower one which may contribute to the increase in the deviation between the computed and measured heights.

Moreover, our method can handle problems with complex geometries and it can be extended to include viscous effects, which are difficult for the BIM approximation. Furthermore, if the interior velocity field or pressure is of interest, BIM approximations require additional calculations.

10 Conclusions

The combination of an adaptive-grid technique and fully-implicit flow solvers for the time-dependent plunging breaker has been developed. Excellent agreement with the experiment validates the current algorithm. The proposed method is so general that can be easily extended to three dimensions and can be applied to transient flow problems with moving boundaries such as offshore structures.

References

[1] Baker, G.R., Meiron, D. I. and Orszag, S. A. *J. of Fluid Mechanics*, vol 123, pp. 477, 1982.

[2] Baker, T.J., "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," *AIAA 8th CFD conference*, pp. 255-270, 1987.

[3] Baker, T. J., "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation," *Engineering with Computers*, Vol. 5, pp. 161–175, 1989.

[4] Baker, T. J., "Triangulations, Mesh Generation and Point Placement Strageties," *Frontiers of Computational Fluid Dynamics 1994*, John Wiley, pp. 61-75, 1994.

[5] Belov, A., Martinelli, L., and Jameson, A., "A Novel Fully Implicit Multigrid Driven Algorithm for Unsteady Incompressible Flow Calculations," to be presented at the Second European Computational Fluid Dynamics Conference, Stuttgart, Germany, September 1994.

[6] Belov, A., Martinelli, L., and Jameson, A., "A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations," AIAA Paper 95-0049, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995

[7] Bowyer, A., "Computing Dirichlet Tesselation ," *the Computer Journal*, vol. 24, no. 2, 162-166, 1981.

[8] Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, pp. 12–26, 1967.

[9] Dole, J.W., "An efficient Surface-Integral Algorithm Applied to Unsteady Gravity Waves ," Journal of Computational Physics, Vol. 103, pp. 90-115, 1992.

[10] Dommermuth, G.D., Yue, D.K.P., *et al.*, "Deep-Water Plunging Breakers: A Comparison Between Potential Theory and Experiments ," Journal of Fluid Mechanics, Vol. 189, pp. 423-442, 1988.

[11] Dreyer, J. J., *Finite Volume Solutions to the Steady Incompressible Euler Equations on Unstructured Triangular Meshes*, Master's Thesis, Department of Mechanical and Aerospace Engineering, Princeton University, June 1990.

[12] Farmer, A., Martinelli, L., and Jameson, A., "A Fast Multigrid Method for Solving Incompressible Hydrodynamic Problems with Free surfaces," AIAA Paper 93-0767, AIAA 31rd Aerospace Sciences Meeting, Reno, Nevada, January 1993.

[13] Harlow, F. H. and Welch, J. E., "Numerical Calculation of Time Dependent Viscous Incompressible Flow of Fluid with Free Surface" , Physics of Fluids, vol. 8, pp. 2182-2189, 1965.

[14] Haussling, H.J. and Van Eseltine, R.T., "Finite-Difference Methods for Transient Potential Flow with Free Surface ," 1st International Conference on Numerical Ship Hydrodynamics , pp.295-313, 1975.

[15] Fedorenko, R.P., "The Speed Convergence of One Iterative Process," *Journal of Computational Physics*, Vol. 39, pp. 201-225,1981.

[16] Hirt, C.W. and Nichols, B.D., "Volume of Fluid (VOF) Method For the Dynamics of Free Boundaries ," Journal of Computational Physics, Vol. 39, pp. 201-225,1981.

[17] Hino, T., Martinelli, L., and Jameson, A., "A Finite-Volume Method with Unstructured Grid for Free Surface Flow Simulations," Sixth International Conference on Numerical Ship Hydrodynamics, Ship Research Institute, Tokyo, Japan, August 1993.

[18] Israeli, M. and Orszag, S.A., "Approximation of Radition Boundary Conditions," Journal of Computational Physics, Vol. 41, pp. 115-135, 1981.

[19] Jameson, A. and Caughey, D. E. , "A finite volume method for transonic potential calculations ," Proc. AIAA Third Computational Fluid Dynamics Conference, Albuquerque, 1977, 33-54.

[20] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, June 1991.

[21] Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," MAE Report 1743, October 1985, also in the Proceedings of the Second European Conference on Multigrid Methods, in Lecture Notes in Mathematics, Vol. 1228, edited by Trottenburg, U., and Hackbusch, W., Springer-Verlag, pp. 166–201, 1986.

[22] Lawson, C. L., "Transforming Triangulations ," Discrete Math, vol. 3, 365-372, 1972.

[23] Lin, P.T., *Implicit Time Dependent Calculations for Compressible and Incompressible Flows on Unstructured Meshes* , Master Thesis, Department of Mechanical and Aerospace Engineering, Princeton University, 1994.

[24] Liou, B.-H., Martinelli, L., and Jameson, A., "A Fully-Implicit Multigrid Driven Algorithm for Computing Time-Resolved Non-linear Free-Surface Flow on Unstructured Grids ," Proceedings of the ASME Fluid Engineering Summer Meeting (1996), vol. 3, pp. 349-354.

[25] Lo, S. H., "A New Mesh generation Scheme for Arbitrary Planar Domains ," Int. J. Numer. Methods Engrg, 21 1403-1426, 1985.

[26] Longuet-Higgins, M. S. and Cokelet, E.D. , "The Deformation of Deep Surface Waves on Water. I. A Numerical Method of Computation ," Proc. R. Soc. Lond. A 350, 1-26, 1976.

[27] Löhner, R "An Adaptive Finite Element Solver For Transient Problems With Moving Bodies ," Computers and Structures, vol. 30, pp303-317, 1988.

[28] Löhner, R., "Adaptive Remeshing For Transient Problems ," Computer Methods In Applied Mechanics and Engineering, vol. 75, pp. 195-214, 1989.

[29] Löhner, R. and Parish, P., "Generation of Three -Dimensional Unstructured Grids by The Advancing Front Methods ," Int. J. Numer. Methods in Fluids, 8, 1135-1149, 1988.

[30] Löhner, R., Morgan, K, Peraire, J, and vahdati, M. , "Finite Element Flux-Corrected Transport(FEM-FCT) for Euler and Navier-Stokes Equations ," Int. J. Numer. Methods in Fluids, 7, 1093-1109, 1987.

[31] Martinelli, L., *Calculations of Viscous Flow with a Multigrid Method*, Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, Princeton University, 1987.

[32] Mavriplis, D., *Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*, Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, Princeton University, 1987.

[33] Melson, N. D., Sanetrik, M. D., and Atkins, H. L., "Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration," Presented at the Sixth Copper Mountain Conference on Multigrid Methods, Copper Mountain, Colorado, April 1993.

[34] Mortenson, M. E., Geometric Modeling, New York, Wiley and Sons, 1985.

[35] Peraire, J, Peiro, J., Formaggia, L., Morgan, K., and Zienkiewicz, O. C. , "Finite Element Euler Computation in Three Dimensions ," Int. J. Numer. Methods in Engrg., 26, 1988.

[36] Mitty, T. J., *Development of a Delaunay-Based Adaption Scheme with Applications to Complex Three-Dimensional Rotational Flows*, Ph.D. Thesis, Department of Mechanical and Aerospace Engineering, Princeton University, January 1993.

[37] Miyaka, T., Sakamoto, Y., Tokunaga, H., and Satofuka, N., "Numerical Solution of Incompressible Flow Using Two-Step, One-Stage Runge-Kutta Time Intergration Scheme ," Presented at the 1st European Computational Fluid Dynamics Conference, Brussels, Belgium, 7-11, September, 1992.

[38] Osher, S. and Sethian, J. A., "Fronts Propagating With Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation. ," *Journal of Computational Physics*, Vol. 79, pp. 12, 1988.

[39] Rizzi, A. and Eriksson, L., "Computation of Inviscid Incompressible Flow with Rotation," *Journal of Fluid Mechanics*, Vol. 153, pp. 275–312, 1985.

[40] Rebay, S., "Efficient Unstructured Mesh generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm ," Journal of Computational physics, Vol. 106, 125-138, 1993.

[41] Vinje, T. and Brevig, P., Adv. Water Resour., 4, 77, 1981.

[42] Watson, D. F., "Computing the n-Dimensional Tessellation with Application to Voronoi Polytopes ," *the Computer Journal*, vol. 24, no. 2, 167-172, 1981.

[43] Welch, J. E., Hawlow, F.H., Shannon, J. P. and Daly, B.J., "The MAC Method: A Computing Technique For Solving Viscous, Incompressible , Transient Fluid Flow Problems Involving Free Surfaces ," *Los Alamos Scientific Laboratory Report*, LA-3425, 1965.

[44] Williams, G. P., "Numerical Integration of the Three-Dimensional Navier-Stokes Equations for Incompressible Flow," *Journal of Fluid Mechanics*, vol. 37, part 4, 727-750.