A Finite-Volume Method with Unstructured Grid for Free Surface Flow Simulations

Takanori Hino
Ship Research Institute
Mitaka Tokyo 181 JAPAN
Luigi Martinelli and Antony Jameson
Princeton University
Princeton, NJ 08544 USA

Presented to the
Sixth International Conference
on
Numerical Ship Hydrodynamics

Ship Research Institute
Tokyo, Japan
August, 1993

A Finite-Volume Method with Unstructured Grid for Free Surface Flow Simulations

Takanori Hino
Ship Research Institute
Mitaka Tokyo 181 JAPAN
Luigi Martinelli and Antony Jameson
Princeton Univercity
Princeton, NJ 08544 USA

ABSTRACT

An unstructured grid method developed initially for the transonic inviscid flow is applied to free surface problems around submerged hydrofoils. The flow domain around a submerged body is divided into triangular cells, which makes up the unstructured grid system fitted to a free surface boundary. The incompressible Euler equations and the continuity equation with artificial compressibility are discretized by the finite-volume method in the unstructured grid. Time integration is made by the Runge-Kutta method. Nonlinear free surface conditions are implemented in the scheme. Several techniques for convergence acceleration are used, including the local time steeping, the residual smoothing and the unstructured multigrid. The outline of numerical procedure is presented together with the results of applications. Comparisons of the results with experimental data prove accuracy and efficiency of the present method.

NOMENCLATURE

CFL	Courant number
$\boldsymbol{D}$	dissipation term
F	Froude number
$\boldsymbol{f}$	flux in x-direction
g	gravitational constant
$\boldsymbol{g}$	flux in y-direction
h	wave height
I_k^{k+1}	interpolation operator in multigrid scheme
L	chord length of hydrofoil
$\boldsymbol{P}$	pre-conditioning matrix
$\boldsymbol{P_k}$	forcing function in multigrid scheme
p	pressure without hydrostatic component
$\hat{p}$	pressure
Q_k^{k+1}	residual transfer operator in multigrid scheme
$\boldsymbol{Q}$	convective term
$\boldsymbol{R}$	residual
s	submergence of hydrofoil
T_k^{k+1}	solution transfer operator in multigrid scheme
t	time
u	velocity in x-direction
v	velocity in y-direction
$\boldsymbol{w}$	solution vector
x	horizontal Cartesian coordinate
y	vertical Cartesian coordinate
α_m	parameters in Runge-Kutta scheme
β^2	artificial compressibilty parameter
β_{q_r}	parameter of Runge-Kutta scheme
$\gamma(x)$	damping term of wave height equation
γ_{q_r}	parameter of Runge-Kutta scheme
ϵ	parameter of residual smoothing
λ	wave speed

INTRODUCTION

Free surface flows have significant importance in ship hydrodynamics. Wave resistance is the major part of the resistance that determines the propulsive performance of ships. Also, waves generated by a ship interact with the boundary layer along a ship hull and affect stern flows which are important for a propeller design. Motions of ships

propulsive performance of ships. Also, waves generated by a ship interact with the boundary layer along a ship hull and affect stern flows which are important for a propeller design. Motions of ships or floating marine structures in ocean waves are of practical importance. Impact loads due to the large ocean waves sometimes damage ships or marine structures. A number of methods have been developed to solve these free surface problems. However, the nonlinearity of the problems makes it difficult to predict the properties of free surface flows accurately and efficiently.

Rapid development of computer hardwares and softwares in recent years enables the large-scale computation. Thus, Computational Fluid Dynamics (CFD) becomes another way to analyze flow properties. CFD activities in ship hydrodynamics have been mainly for the prediction of viscous flows around a ship stern[1,2] in which free surface is treated as a symmetric boundary. Free surface flows have been treated by a kind of a panel method assuming inviscid flows[3]. However, because there are interactions between viscous flows and free surface waves, it is desirable to solve viscous flow problems under free surface effects. Attempts to this direction are Hino[4], Miyata et al.[5], Tahara et al.[6] and so on.

When one solves nonlinear free surface flows around a ship with a boundary-fitted grid, which is common in the recent CFD method, a grid must be generated at each time step, because free surface is dynamic in time. The grid generation is not an easy task even without a free surface movement when the body geometry becomes complex. Free surface deformations which are large particularly near the body gives additional complexity to the grid generation.

CFD in aerodynamics is much older than its counterpart in ship hydrodynamics. Various new technologies have been invented in the CFD for aerodynamics. Among them, Jameson et al.[7] developed unstructured grid method for transonic flow computations which uses the triangular grid rather than rectilinear grid in the structured grid case. Later, this method is applied to incompressible flow problems by introducing the artificial compressibility[8]. This unstructured grid method has capability to cope with the geometrical complexity and therefore suitable for free surface flow problems.

In this paper, an unstructured grid method is applied to free surface problems. The problems concerned are flows around a submerged hydrofoils. This is chosen partly because the problem is much simpler compared with flows around a surface-piercing body and partly because the original method is for transonic aerofoils and can be naturally extended to hydrofoil problems.

The governing equations are incompressible Euler equations. Though the final goal of the study is the viscous flow computations, the Euler equations are selected as the governing equation for its simplicity. Artificial compressibility is introduced in the continuity equation. This makes the system of equations hyperbolic and the well-developed efficient techniques to solve hyperbolic equations can be used.

NUMERICAL PROCEDURES

Governing Equations

Governing equations are two-dimensional incompressible Euler equations and are expressed in the form non-dimensionalized by the chord length of a hydrofoil L, the uniform flow velocity U and the fluid density ρ as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2 + \hat{p})}{\partial x} + \frac{\partial (uv)}{\partial y} = 0 \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2 + \hat{p})}{\partial y} + \frac{1}{F^2} = 0 \tag{3}$$

where (x, y) are Cartesian coordinates (y is upward positive) and (u, v) are the velocity components in (x, y) directions, respectively. $\hat{p}$ is the static pressure and t is time. F is Froude number defined using the gravitational constant g as

$$F = \frac{U}{\sqrt{gL}} \tag{4}$$

Since there is no term associated with the time derivative of pressure in the governing equations (1)-(3), difficulties come out when one solves these equations in the time-marching manner. Usually pressure field is computed by the Poisson equation which is derived from the divergence of the momentum equations (2)-(3) and the continuity equation (1) in such a way that the velocity field satisfies the continuity condition at each time step [9].

When only the steady state solution is required, the alternative approach called the artificial compressibility method can be used. In this

method first proposed by Chorin [10], the continuity equation is modified by introducing the pseudo-compressibility as follows:

$$\frac{\partial p}{\partial t} + \beta^2(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) = 0 \qquad (5)$$

where β^2 is the artificial compressibility parameter. When the solution becomes steady, the equation (5) recovers the original form (3). Since the system of equations (5), (2) and (3) is hyperbolic, efficient numerical solution methods for the hyperbolic equations can be applied.

The parameter β^2 is determined by using the local velocity magnitude as

$$\beta^2 = r_b * \min(u^2 + v^2, \beta^2_{\min}) \qquad (6)$$

where r_b is a global constant and the parameter $\beta^2_{\min}$ is used to prevent β^2 from approaching zero near the stagnation point. $r_b = 5$ and $\beta^2_{\min} = 0.3$ are typical values in the present study.

Eqs.(5), (2) and (3) can be rewritten in the vector form as

$$\frac{\partial \boldsymbol{w}}{\partial t} + \boldsymbol{P}(\frac{\partial \boldsymbol{f}}{\partial x} + \frac{\partial \boldsymbol{g}}{\partial y}) = 0 \qquad (7)$$

where

$$\boldsymbol{w} = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} u \\ u^2 + p \\ uv \end{bmatrix}, \boldsymbol{g} = \begin{bmatrix} v \\ uv \\ v^2 + p \end{bmatrix} \qquad (8)$$

and p is the pressure without the hydrostatic component, i.e.,

$$p = \hat{p} + \frac{y}{F^2} \qquad (9)$$

$\boldsymbol{P}$ is a matrix defined as

$$\boldsymbol{P} = \begin{bmatrix} \beta^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (10)$$

Boundary conditions needed for free surface flow problems are a body surface condition, a free surface condition and a far field condition. The first one is the free-slip condition in case of the inviscid flow, that is,

$$u\, n_x + v\, n_y = 0 \qquad (11)$$

where (n_x, n_y) are the unit vector outward normal to the body surface.

The second one, the free surface condition consists of two conditions. One is the dynamic condition that states the continuity of stresses on the air-liquid interface. For the inviscid case this is expressed as

$$\hat{p} = p_0 \quad \text{on} \quad y = h \qquad (12)$$

or, equivalently,

$$p = p_0 + \frac{h}{F^2} \quad \text{on} \quad y = h \qquad (13)$$

where p_0 is atmospheric pressure (assumed to be constant) and $y = h(x;t)$ is the free surface location. The other free surface condition is called the kinematic condition that means the fluid particles on the free surface keep remaining on it. This is written as

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} - v = 0 \qquad (14)$$

The free surface shape can be updated by Eq.(14) in the time-marching manner.

The far field conditions are as follows. At far upstream, flow is uniform and free surface is undisturbed. On the other hand, the waves generated by a hydrofoil propagate to far down stream. Water depth is assumed infinite.

Spatial Discretization

Finite-Volume Method

In a finite-volume formulation a solution domain is divided into small cells. The present method employs unstructured grid in which every cell is triangular. One of the superiorities of unstructured grids over structured ones is its flexibility to deal with complex geometry. Therefore, unstructured grids are particularly suitable to free surface flow problems in which the deformation of free surface boundary causes further geometrical complexity in addition to that of body configuration.

The flow variables (u, v) and p are defined at the vertices of each triangle. The control volume for a given node i is taken as the union of all the triangles which share that node as a vertex as shown in Fig.1. The integration of the governing equation (7) over this control volume yields

$$\begin{aligned} \frac{\partial}{\partial t}\iint_\Omega \boldsymbol{w}\, dx\, dy &= -\iint_\Omega \boldsymbol{P}(\frac{\partial \boldsymbol{f}}{\partial x} + \frac{\partial \boldsymbol{g}}{\partial y})\, dx\, dy \\ &= -\oint_{\partial\Omega} \boldsymbol{P}(\boldsymbol{f}\, dy - \boldsymbol{g}\, dx) \end{aligned} \qquad (15)$$

where Ω means the control volume and $\partial\Omega$ is its boundary. Since the grid is aligned to the free

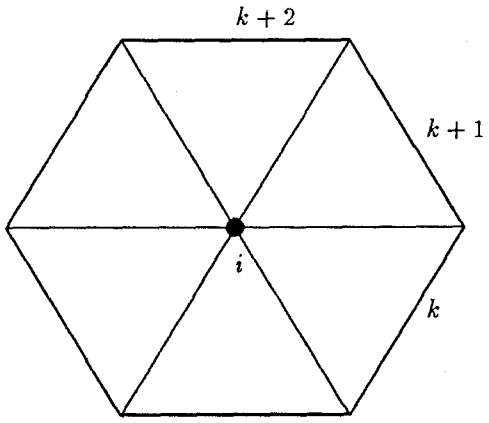

Figure 1: Control volume around the node i

surface boundary which moves in time, the grid is time-dependent and the effects of grid movement must be taken into account for the time-accurate computation. However, since the present method employs the steady state formulation and the transient solution does not have physical meaning, one can make approximation to drop all the terms associated with the grid movement. Thus, $\boldsymbol{f}$ and $\boldsymbol{g}$ in Eq.(15) have the same form as in Eq.(8). In the discrete form, Eq.(15) becomes

$$\frac{d}{dt}(S_i\,\boldsymbol{w}_i) + \sum_{k=1}^{n_e} \boldsymbol{P}(\boldsymbol{f}_k\,\Delta y_k - \boldsymbol{g}_k\,\Delta x_k) = 0 \quad (16)$$

where S_i is the area of the control volume around the node i which is computed by the summation of the area of each triangle in the control volume. The summation in Eq.(16) is taken over all the edges surrounding the control volume and n_e is the number of the edges. Also, $(\Delta y_k, -\Delta x_k)$ gives the unscaled outward normal vector of the k-th edge. $\boldsymbol{f}_k$ and $\boldsymbol{g}_k$ are the flux vectors evaluated by taking average of the values at both ends of the edge. This discretization corresponds to the central difference scheme in the structured grid case. The time integration scheme for Eq.(16) is described in the subsequent section.

Artificial Dissipation

Since the evaluation of Eq.(16) described above is the scheme equivalent to the central difference scheme for the Euler equations, this scheme is not stable due to the decoupling of neighboring node unless one adds the artificial dissipation terms to the equations. To keep the second order accuracy of the scheme, the fourth-order dissipation models are used in the scheme, while the second-order dissipation terms used in the compressible flow code [11] to prevent oscillation near shocks are not used.

By adding the artificial dissipation terms, Eq.(16) is rewritten as

$$\frac{d}{dt}(S_i\,\boldsymbol{w}_i) + \boldsymbol{Q}_i(\boldsymbol{w}) - \boldsymbol{D}_i(\boldsymbol{w}) = 0 \quad (17)$$

where

$$\boldsymbol{Q}_i(\boldsymbol{w}) = \sum_{k=1}^{n_e} \boldsymbol{P}(\boldsymbol{f}_k\,\Delta y_k - \boldsymbol{g}_k\,\Delta x_k) \quad (18)$$

and $\boldsymbol{D}_i(\boldsymbol{w})$ is the dissipation terms. The dissipation terms are evaluated as follows. First, the undivided Laplacian in the computational space is approximated as,

$$\nabla^2 \boldsymbol{w}_i = \sum_{j=1}^{n_n} (\boldsymbol{w}_j - \boldsymbol{w}_i) \quad (19)$$

The dissipation terms are constructed by using this $\nabla^2 \boldsymbol{w}$ as

$$D(\boldsymbol{w})_i = \sum_{j=1}^{n_n} \epsilon \lambda_{ij} (\nabla^2 \boldsymbol{w}_j - \nabla^2 \boldsymbol{w}_i) \quad (20)$$

where ϵ is a global constant which controls the amount of dissipation and λ_{ij} is a scale factor. The summation is taken over all the nodes on the boundary of the control volume around the node i and n_n is the number of the nodes.

From the analogy to upwind differencing, the scale factor λ_{ij} is determined as follows. First, the maximum wave speed is determined by the spectral radii of the flux Jacobian matrices as

$$\lambda = \rho(\boldsymbol{A}\Delta y - \boldsymbol{B}\Delta x) \quad (21)$$

where

$$\boldsymbol{A} = \boldsymbol{P}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}} = \begin{bmatrix} 0 & \beta^2 & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{bmatrix},$$
$$\boldsymbol{B} = \boldsymbol{P}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{w}} = \begin{bmatrix} 0 & 0 & \beta^2 \\ 0 & v & u \\ 1 & 0 & 2v \end{bmatrix} \quad (22)$$

This yields

$$\lambda = |(u\,\Delta y - v\,\Delta x)| + \sqrt{(u\,\Delta y - v\,\Delta x)^2 + \beta^2(\Delta x^2 + \Delta y^2)} \quad (23)$$

Thus, a scale factor λ_{ij} which is λ associated with the edge consists of the nodes i and j is defined as

$$\lambda_{ij} = \max(|q_i|, |q_j|) + \sqrt{\max(c_i, c_j)} \qquad (24)$$

where

$$q_i = u_i \Delta y_{ij} - v_i \Delta x_{ij} \qquad (25)$$

$$c_i = q_i^2 + \beta_i^2 \sqrt{\Delta x_{ij}^2 + \Delta y_{ij}^2} \qquad (26)$$

and $(\Delta x_{ij}, \Delta y_{ij})$ is the vector from the node i to the node j. q_j and c_j can be evaluated by replacing u_i, v_i and β_i^2 in the above equations by u_j, v_j and β_j^2, respectively.

Boundary Conditions

Body Boundary Condition

Free-slip condition on the body (11) is implemented in the following way. To determine two velocity components (u, v) on the boundary, two conditions are required. One condition is apparently the free slip condition (11). The other condition is that the tangential velocity does not have gradient in the normal direction, that is,

$$\frac{\partial q_t}{\partial n} = 0 \qquad (27)$$

where n is the normal direction and q_t is the tangential velocity defined by using the unit outward vector on the body (n_x, n_y) as

$$q_t = u\, n_y - v\, n_x \qquad (28)$$

For the node which lies on the body boundary, the tangential velocity is extrapolated from the inside in such a way that Eq. (27) is satisfied. First, for each node on the body boundary, the edge from which velocity is extrapolated is searched. Searching procedure is 1) search the triangle that consists of the boundary node under consideration and two internal nodes, 2) the edge formed by the two internal nodes is registered as a candidate edge, 3) from the candidates select the edge in such a way that the angle between the vector from the boundary node to the mid-point of the edge and the outward normal vector of the boundary node is minimum. Thus, the velocity can be extrapolated from the direction approximately normal to the body surface. Suppose that the boundary node is denoted as O and that the end points of the corresponding edge are A and B as shown in Fig. 2, the extrapolation formula is

$$(q_t)_o = (1 - \kappa)(q_t)_A + \kappa (q_t)_B \qquad (29)$$

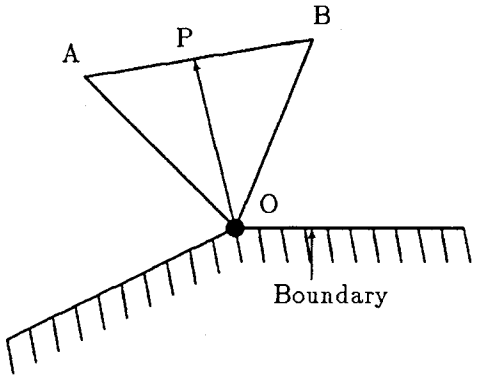

Figure 2: Velocity extrapolation of the boundary node

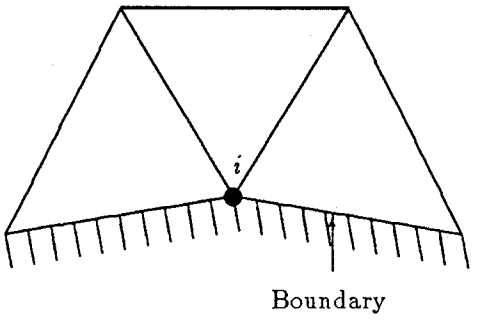

Figure 3: Control volume for the boundary node

where

$$\kappa = \frac{\overline{AP}}{\overline{AB}} \qquad (30)$$

where P is the intersection point between $\overline{AB}$ and the vector normal to the boundary (see Fig. 2). From Eqs.(29) and (11), the velocity component on the body is computed as

$$u = (q_t)_o\, n_y,\ v = -(q_t)_o\, n_x \qquad (31)$$

The pressure on the body is computed by the modified continuity equation (5). The control volume is taken as shown in Fig. 3. Mass fluxes across the boundary edges are set zero because of the free-slip condition. The discretization is carried out in the usual way except that the node is on the perimeter of the control volume.

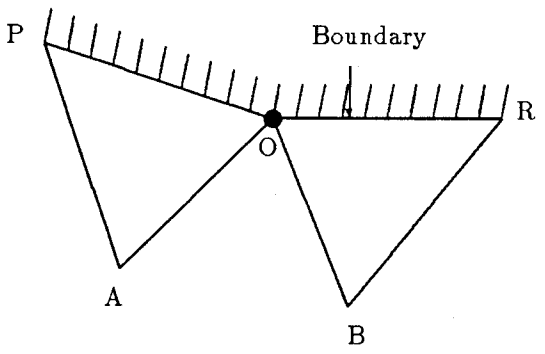

Figure 4: Velocity extrapolation on the free surface

Free Surface Condition

Since the the grid is aligned to the free surface boundary, the free surface dynamic condition (13) is satisfied by simply setting pressure value on the free surface to $p_0 + h/F^2$.

The velocity on the free surface is extrapolated from inside in such a way that the velocity gradient in the normal direction is zero. Though this can be approximated in the same way as for the body boundary, the simpler extrapolation scheme is preferable because re-calculation of extrapolation coefficient κ at each time-step due to the grid movement is time-consuming. (Note that the grid points close to the body boundary do not move in time as described in the subsequent section.) Suppose that the free surface node O is the common node for the boundary edge $\overline{\mathrm{PO}}$ and $\overline{\mathrm{OR}}$ and these edges are the part of the triangles $\triangle$POA and $\triangle$OBR, respectively as depicted in Fig.4 (Note that node O, A and B do not necessarily form a triangle), then velocity at node O is computed by

$$u_O = \frac{1}{2}(u_A + u_B),\ v_O = \frac{1}{2}(v_A + v_B) \qquad (32)$$

In the above approximation, the mid-point of $\overline{\mathrm{AB}}$ is assumed to be in the normal direction from the node O.

The kinematic condition (14) is used to update the free surface shape. The spatial discretization is based on the third-order upwind finite-differencing scheme and the equation for the node i becomes

$$\frac{dh_i}{dt} + u_i \frac{2h_{i+1} + 3h_i - 6h_{i-1} + h_{i-2}}{2x_{i+1} + 3x_i - 6x_{i-1} + x_{i-2}} - v_i = 0 \qquad (33)$$

where the node numbering is sequential from upstream to downstream and (u_i, v_i) is the velocity at the node i whose coordinate is given by (x_i, h_i).

Far Field Conditions

At inflow, flow is uniform, that is,

$$u = 1,\ v = 0,\ p = 0,\ h = 0 \qquad (34)$$

are given.

At outflow, since waves generated by a hydrofoil propagate to infinite downstream, flow is not uniform. To prevent the reflection of waves to the solution domain, the outflow conditions must be carefully implemented. The open boundary conditions for free surface problems are treated by the various methods [12]. Among them, the procedure used here is the artificial damping method, in which waves going through the outflow boundary is dissipated by adding artificial damping terms in the flow equation.

The free surface kinematic condition is modified as

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} - v + \gamma(x) = 0 \qquad (35)$$

where γ is the artificial damping terms defined as

$$\gamma(x) = \begin{cases} A\left(\frac{x - x_d}{x_o - x_d}\right)^2 h(x) & \text{if } x_d \le x \le x_o \\ 0 & \text{otherwise} \end{cases} \qquad (36)$$

where A is a constant that controls the amount of damping and x_o is the x-coodinate of the outflow boundary. x_d is defined as

$$x_d = x_o - 2\pi F^2 \qquad (37)$$

That is, the damping zone is set one wavelength computed by the linear theory from the outflow boundary. The quadratic form of the damping term in Eq.(36) gives the gradual increase of dissipation, which prevents the reflection of waves at the edge of the damping zone.

Velocity and pressure on the outflow boundary is computed by the control volume of Fig.3, which is equivalentto rhe one-sided differencing.

At the bottom boundary, pressure p is assumed to zero, which corresponds to the hydrostatic value and velocity is computed from the momentum equations with the one-sided differencing using the control volume of Fig.3.

Time Stepping

As the time integration scheme, the explicit multi-stage Runge-Kutta scheme originally developed for compressible flows [11] is used here.

As stated earlier, the grid is time-dependent in the present case, therefore, the control volume is also time-dependent. However, when only steady state solutions are of interest, one can simplify the solution procedure by dropping the terms associated with the grid movement. Suppose that one has the grid and the solution at time step (n), the procedure to proceed one time step is as follows. The flow equations (17) are solved assuming that the grid does not move in time, that is, S_i ,Δx and Δy appeared in Eq.(17) are evaluated using the grid at time step (n) and are kept constant in time. Thus, Eq.(17) can be rewritten as

$$\frac{d\boldsymbol{w}_i}{dt} + \frac{1}{S_i}(\boldsymbol{Q}_i(\boldsymbol{w}) - \boldsymbol{D}_i(\boldsymbol{w})) = 0 \qquad (38)$$

This equation gives an approximated solution at time step $(n+1)$. Then, the free surface kinematic condition (14) is solved in the same manner as the flow equations and the wave height at time step $(n+1)$ is obtained. Next, the grid points are redistributed in such a way that the grid is conformed to the newly computed free surface configuration. During this redistribution process, the number of grid points and the edge connectivity are not changed so as to avoid the re-triangulation at each time step. The flow variables at time step $(n+1)$ are set equal to the values computed under the approximation that the grid is fixed. The geometric quantities S_i, Δx and Δy are recomputed by using the new coordinates and the computation proceeds to the next time-step.

Since the grid points do not move in time when a solution becomes steady, this approximated procedure must give the same steady state solutions as the time-accurate scheme does.

Time integration scheme for the flow equation (38) and the wave height equation (14) is the Runge-Kutta method which is a class of one-step multi-stage explicit schemes. The general m-stage solution procedure for Eq.(38) from the time step (p) to $(p+1)$ can be written as follows:

$$\boldsymbol{w}^{(0)} = \boldsymbol{w}^p \qquad (39)$$

$$\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} - \alpha_1 \frac{\Delta t}{S}\boldsymbol{R}^{(0)}(\boldsymbol{w}) \qquad (40)$$

$$\boldsymbol{w}^{(2)} = \boldsymbol{w}^{(0)} - \alpha_2 \frac{\Delta t}{S}\boldsymbol{R}^{(1)}(\boldsymbol{w}) \qquad (41)$$

$$\cdots$$

$$\boldsymbol{w}^{(m)} = \boldsymbol{w}^{(0)} - \alpha_m \frac{\Delta t}{S}\boldsymbol{R}^{(m-1)}(\boldsymbol{w}) \qquad (42)$$

$$\boldsymbol{w}^{p+1} = \boldsymbol{w}^{(m)} \qquad (43)$$

where $\boldsymbol{R}^{(q)}(\boldsymbol{w})$ is the residual evaluated at q-th stage and is defined by the weighted average of the residuals computed by the flow variables of previous stages, i.e.,

$$\boldsymbol{R}^{(q)}(\boldsymbol{w}) = \sum_{r=0}^{q}\{\beta_{qr}\boldsymbol{Q}(\boldsymbol{w}^{(r)}) - \gamma_{qr}\boldsymbol{D}(\boldsymbol{w}^{(r)})\} \qquad (44)$$

α_m, β_{qr} and γ_{qr} define the particular scheme. The values of these coefficients for the 4-stage scheme used in this study are as follows.

$$\alpha_1 = 1/3,\ \alpha_2 = 4/15,\ \alpha_3 = 5/9,\ \alpha_4 = 1 \qquad (45)$$

$$\beta_{qr} = \begin{cases} 1 & \text{if } q = r \\ 0 & \text{otherwise} \end{cases} \qquad (46)$$

$$\begin{aligned} &\gamma_{00} = 1 \\ &\gamma_{10} = 0.5, \quad \gamma_{11} = 0.5 \\ &\gamma_{20} = 0.5, \quad \gamma_{21} = 0.5, \quad \gamma_{22} = 0 \\ &\gamma_{30} = 0.5, \quad \gamma_{31} = 0.5, \quad \gamma_{32} = 0, \quad \gamma_{33} = 0 \end{aligned} \qquad (47)$$

Thus, the dissipation terms are evaluated twice in one time-step.

The free surface kinematic condition (14) is solved in the same manner except that there are no dissipation terms due to the use of upwind differencing.

Grid Generation and Grid Movement

The generation of unstructured triangular grid around a body can be achieved by various ways. Among them, the Delaunay triangulation method [14] and the advancing front method [13] are commonly used in the CFD field. The former is the procedure to establish unique triangulation of given grid points covering solution domain, while the latter is the method to generate points and connect them simultaneously.

The unstructured grid around the submerged hydrofoil used here is generated by the Delaunay triangulation. The set of points are generated by the combination of the conformal mapping around the region close to a foil and the algebraic method in the other region. The conformal mapping is an established way to generate O-grid around a foil of an arbitrary shape. The algebraic method is required since the outer

boundary is rectangular and since the grid spacing control is needed for the better resolution of the free suface region. The points are clustered in the region above and behind the body and near the free surface where the free surface deformation is expected to be large.

The grid movement is carried out in the following way. The initial grid is generated by assuming the flat free surface. On this stage y-coordinate of the uppermost point generated by the conformal mapping is searched and stored as y_u. y_u is the uppermost extent of the 'inner grid' that is generated in the region close to the body and only the grid point above y_u are allowed to move following free surface movement. Assume that the wave height at time step n is given by $h^n(x)$, then the grid movement of the node i is defined as

$$x_i^{n+1} = x_i^n \tag{48}$$

$$y_i^{n+1} = \begin{cases} y_u + \dfrac{(h^{n+1}(x_i) - y_u)}{(h^n(x_i) - y_u)}(y_i^n - y_u) & \text{if } y_i^n > y_u \\ y_i^n & \text{otherwise} \end{cases} \tag{49}$$

where $h(x_i)$ is the wave height at $x = x_i$ and is computed by the linear interpolation because x_i does not necessarily coincide with the x-coordinate of the free surface nodes. Thus, all the points above y_u move vertically due to the free surface movement. The moving distance is linearly distributed between h^{n+1} and y_u. By this procedure the grid points near the body do not move and the complicated re-distribution procedure for points near the body is avoided.

Convergence Acceleration Techniques

Three techniques are used to accelerate the convergence of solutions to the steady state in the present scheme. A local time stepping is the method in which the solution at each point proceeds in time with the time step defined locally from the local stability limit, while a residual smoothing is used to increase the bound of the stability limit of the time stepping scheme itself. A multigrid method is an efficient way to accelerate the convergence, where the time stepping is carried out by using successively coarser grids as well as the original finest grid.

Local Time Step

For explicit schemes the maximum permissible time step is limited by the Courant-Friedrichs-Lewy (CFL) condition. In one dimensional case, this is written as

$$\Delta t \leq CFL\frac{\Delta x}{c} \tag{50}$$

where CFL is the maximum Courant number permitted by the scheme, Δx is the grid size, and c is the maximum wave speed. When one uses the globally constant time step, Δt must be smaller than the minimum value of $CFL\Delta x/c$.

In practice, the grid spacing is not uniform due to the clustering of points. Therefore the time step is determined based on the minimum grid spacing. This yields the small time step and causes slow convergence.

If only steady state solutions are of interest, one can use the locally varying time step at the expense of time-accuracy. The local time step Δt_i is taken as its maximum permissible value, i.e.,

$$\Delta t_i = CFL\frac{\Delta x_i}{c_i} \tag{51}$$

In case of unstructured grid employed here, the above equation is modified as

$$\Delta t_i = CFL\frac{S_i}{\lambda_i} \tag{52}$$

where S_i is the area of the control volume around the node i.

Residual Smoothing

As stated earlier, explicit schemes have the CFL limit of stability. Residual smoothing procedure described below is the way to increase the stability bound of a time stepping scheme. Thus, larger time step can be taken and the fast convergence is achieved. In the method, the residual at the node i, $\boldsymbol{R}_i(\boldsymbol{w})$ is replaced by implicitly averaged value $\tilde{\boldsymbol{R}}_i(\boldsymbol{w})$, where

$$\tilde{\boldsymbol{R}}_i = \boldsymbol{R}_i + \epsilon\nabla^2\tilde{\boldsymbol{R}}_i \tag{53}$$

where ϵ is a constant and the operator ∇^2 is the undivided Laplacian in the computational space defined in Eq.(19). The resulting linear equation

$$(1 - \epsilon\nabla^2)\tilde{\boldsymbol{R}}_i = \boldsymbol{R}_i \tag{54}$$

is solved iteratively by the Jacobi method. This gives the implicit property to the scheme and the CFL limit can be larger than the unsmoothed case. One dimensional analysis shows that one can take arbitrary large time step as far as ϵ is taken correspondingly large [11]. In this study ϵ is taken 0.5.

Unstructured Multigrid

Multigrid method is known as the efficient way to get fast convergence. The concept of the multigrid time stepping applied to the solution of hyperbolic equations by Jameson [11] is to compute corrections to the solution on a fine grid by the time-stepping on a coarser grid.

The general procedure of the multigrid method is as follows. Equations to be solved is written as

$$\frac{d\boldsymbol{w}}{dt} = -\boldsymbol{R}(\boldsymbol{w}) \tag{55}$$

and the subscript k refers as the grid index.

First, the solution $\boldsymbol{w}_k$ is obtained in the fine grid (k) by solving

$$\frac{d\boldsymbol{w}_k}{dt} = \boldsymbol{R}_k(\boldsymbol{w}_k) \tag{56}$$

by the Runge-Kutta scheme described above. Then, the solution is transferred from the fine grid (k) to the next coarser grid $(k+1)$ by

$$\boldsymbol{w}_{k+1}^{(0)} = T_k^{k+1}\boldsymbol{w}_k \tag{57}$$

where T_k^{k+1} is a transfer operator. The solution in the coarse grid is updated by solving the equation

$$\frac{d\boldsymbol{w}_{k+1}}{dt} = -\boldsymbol{R}_{k+1}(\boldsymbol{w}_{k+1}) - \boldsymbol{P}_{k+1} \tag{58}$$

with the Runge-Kutta scheme and $\boldsymbol{w}_{k+1}^{+}$ is obtained. $\boldsymbol{P}_{k+1}$ in the above equation is the forcing function in the coarse grid $(k+1)$ defined as

$$\boldsymbol{P}_{k+1} = Q_k^{k+1}\boldsymbol{R}_k(\boldsymbol{w}_k) - \boldsymbol{R}_{k+1}(\boldsymbol{w}_{k+1}^{(0)}) \tag{59}$$

where Q_k^{k+1} is another transfer operator. The first term of the right-hand-side of Eq.([?]) is the residual transferred from the finer grid and the second term is the residual evaluated by the transferred solution. This second term cancels the residuals in the coarse grid only and the driving force comes from only the residual transfered from the finer grid.

$\boldsymbol{w}_{k+1}^{+} - \boldsymbol{w}_{k+1}^{(0)}$ gives the correction of the solution at the grid $(k+1)$. This procedure is repeated on successively coarser grid. Finally, after the computation of the correction at the coarsest gird, the correction is transferred back from the coarse grid $(k+1)$ to the fine grid (k) by

$$\boldsymbol{w}_k^{+} = \boldsymbol{w}_k + I_{k+1}^{k}(\boldsymbol{w}_{k+1}^{+} - \boldsymbol{w}_{k+1}^{(0)}) \tag{60}$$

where I_{k+1}^{k} is an interpolation operator. The multigrid cycle employed here is V-cycle in which the equations are solved only when the solution moves from the fine grid to the coarse one and the interpolation is used in the transfer of correction from the coarse grid to the fine one.

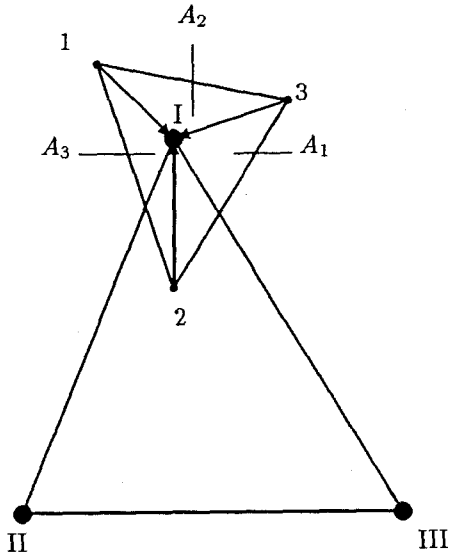

Figure 5: Transfer of solution

In case of the structured grids, the generation of successively coarser grids can be done simply by deleting the alternate points along each grid line. This also makes it easy to define the operators described above. In the unstructured grids, however, neither the grid generation nor the definition of the operators is straightforward.

In the present study, the multigrid strategy of Mavriplis [14] is employed. That is, to keep flexibility of unstructured grids as much as possible, the series of coarser grids are generated independently. The grid generation procedure is repeated fo each grid with changing the grid density parameters.

The operators for the transfers are defined as follows. T_k^{k+1} is the operator with which the solution $\boldsymbol{w}$ transfers from the fine grid to the coarse one. This is defined as

$$\boldsymbol{w}_I = \frac{(A_1\boldsymbol{w}_1 + A_2\boldsymbol{w}_2 + A_3\boldsymbol{w}_3)}{A_1 + A_2 + A_3} \tag{61}$$

where I is the node in the coarse grid under concideraton and 1, 2 and 3 are the nodes forming the triangle in the fine grid which contain the node I. A_1 is the area of the triangle consists of I, 2 and 3 as shown in Fig.5. A_2 and A_3 are defined similarly.

Q_k^{k+1} is the transfer operator of the residuals. The residual at the node I in the coarse grid

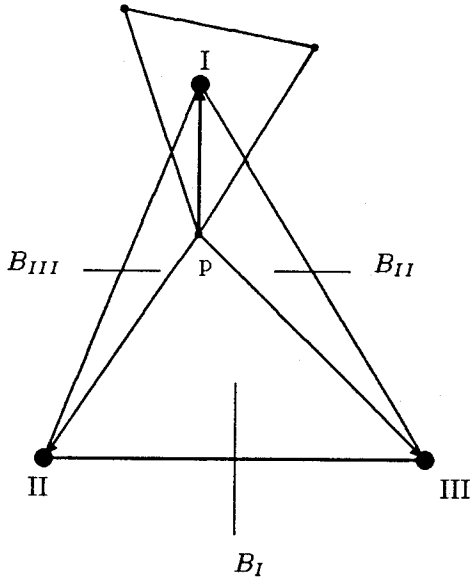

Figure 6: Transfer of residuals

is computed by

$$\boldsymbol{R}_I = \sum_{\Delta}(\sum_{p} \frac{B_I \boldsymbol{R}_p}{B_I + B_{II} + B_{III}}) \qquad (62)$$

The first summation is over the triangles of coarse grid that share the node I as a common vertex. The second summation is taken over all the nodes in the fine grid that lie inside the triangles determined in the first summation. B_I is the area of the triangle consists of the nodes p, II and III as shown in Fig.6. B_{II} and B_{III} are defined similarly. Thus, the residuals in the coarse grid are linearly distributed to the nodes in the fine grid. Note that this procedure is conservative, that means the grand sum of the residuals in the coarse grid is equal to the grand sum of the residuals in the fine grid. This feature is important, because the the driving force in the coarse grid comes only from the fine grid residuals due to the forcing function (59).

Finally, I_k^{k+1} is for the interpolation of the corrections in the coarse grid to the fine grid. From Eq.(60) the correction $d\boldsymbol{w}$ is defined by

$$d\boldsymbol{w} = \boldsymbol{w}^+ - \boldsymbol{w}^{(0)} \qquad (63)$$

The transfer of the correction from the coarse grid to the fine grid is done as follows.

$$dw_1 = \frac{B_I dw_I + B_{II} dw_{II} + B_{III} dw_{III}}{B_I + B_{II} + B_{III}} \qquad (64)$$

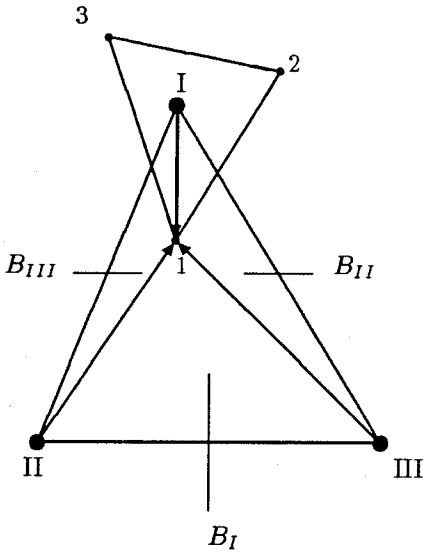

Figure 7: Transfer of correction

where the node 1 is the node of the fine grid and it lies inside the triangle of the nodes I, II and III in the coarse grid as shown in Fig.7. B_I is the area of the triangle of 1, II and III. B_{II} and B_{III} are defined similarly

Since the grid is dynamic in time for the present case, the above operators must be recomputed at each time step using new coordinates of the grid. The most time-consuming part of this re-computation is the search of the triangle in which the certain point lies. The procedure taken here is based on the tree-search algorithm similar to the one used by Mavriplis[14]. First, every triangle has a list of three neighbour triangles. This list does not change by the grid movement because the point connectivity is fixed. At the beginning, the search of the triangle is carried out by the simple search consists of the loop over all the triangles.

The search of triangles after the grid movement are achieved by the following procedure. The search starts with the triangle in which the point under consideration lies before the grid movement. If the point still lies in this triangle, the search ends. If not, the three triangles which are the neighbours of the first triangle are searched next. The search region is extended successively to the neigbours of the neigbours until the desired triangle is found. Since the moving distance of grid points are not large, this search converges usually in a few extensions. The computational time required is much less than that

of the loop over all the triangles.

RESULTS

The numerical procedure described above is applied to free surface flow simulations around a submerged hydrofoil. A hydrofoil section used here is NACA0012 and the angle of attack is set 5 degrees. The depth of submergence s measured at the mid-chord and nondimensionalized by the chord length varies from 1.034 to 0.911. Froude number $F = 0.5672$ and water depth is assumed to be infinite. These conditions correspond with the experiments by Duncan [15] except that the experiments were carried out in the tank of a finite water depth (varies from 1.90 to 1.77, nondimensionalized by a chord length).

For all the computation, the initial computational domain is taken as

$$-7 \leq x \leq 6.25, \ -7 \leq y \leq 0 \qquad (65)$$

unless otherwise noted. $x = 0$ is the 1/4 chord aft from the leading edge of the hydrofoil and $y = 0$ is the undisturbed water level.

Three levels of multigrids used for the computation with the submergence $s = 1.034$ are shown in Fig.8. The finest grid consists of 6,587 nodes and 12,854 triangles. Among them, 64 nodes are distributed on the body. The medium grid has 1,642 nodes and 3,130 triangles while the coarsest one has 409 nodes and 745 triangles. Since these grids are for the converged solution, the free surface configurations correspond to the developed wave field, though the initial grids are generated under the undisturbed free surface. Maginified view of the finest grid is depicted in Fig.9.

The computed pressure distributions with the multi- and single- grid cases are shown in Figs.10 and 11. Both are the results at 400 time cycles with the Courant number CFL being 5.0.

The single grid case shows the undeveloped wave field in which the waves generated by the hydofoil do not yet reach the outflow boundary, while in the multigrid case the converged solution with developed wave field (except for the region close to the outflow boundary where numerical damping is added) is obtained. This is due to the fact that time advancement in one multigrid cycle in the three-level mutigrid is approximately $\Delta t + 2\Delta t + 4\Delta t = 7\Delta t$, where Δt is the time step in the finest grid, because the local time stepping is taken proportional to the cell area. On the other hand, time advancement in the single-grid case in one cycle is just Δt.

For the practical computations shown hereafter the full multigrid scheme is employed, in which the initial solution is obtained with only the coarsest grid, then the solution is transferred to the next finer grid and the solution is updated by the multigrid method using the coarsest grid and the next finer grid. The procedure is repeated with adding one level of multigrid at a time until the finest grid is reached. The cycles for each stage are 1,000 cycles with the single coarsest grid, 500 cycles with the double grid (the medium and the coarsest) and 400 cycles with triple grid (the finest, the medium and the coarsest grids). With this scheme, the residual in the final stage is $O(10^{-4})$.

The effect of numerical damping expressed by Eq.(36) is verified next. Fig.12 compares two computed wave configurations. One has the solution domain of $-7 \leq x \leq 6.25$, and the other has the longer domain of $-7 \leq x \leq 8.25$. The number of grid points in the longer domain case increases in such a way that the grid density is approximately the same in both cases. Since Froude number is 0.5672, the length of the damping zone defined in Eq.(37) is about 2.02. Waves going through the outflow boundary are damped effectively in the damping zone and the reflection of waves on the outflow boundary cannot be observed. Also, the comparison of the long-domain solution and the short-domain one shows that the artificial damping does not affect the flow field upstream of the damping zone.

In Fig.13 the computed wave height is compared with the experimental data by Duncan [15]. The present result is in good agreement with the experimental data. Fig.14 and 15 show the pressure distribution and the velocity vectors in the vicinity of the hydrofoil, respectively. Fig.16 shows the computed Cp distribution on the body.

Fig.17 is the result with $s = 0.951$. Again, the computed wave profile shows good agreement with experimental data [15]. When the submergence of the hydrofoil decreases, the wave amplitude becomes larger and wave length becomes shorter than the deep submergence case. These nonlinear features of wave formations in the experiment are clearly captured by the present method. Note that in the experiment with $s = 0.951$, two types of wave profiles, one is breaking and the other is non-breaking, are obtained depending on the experimental condition. However, the present computation predicts only the non-breaking waves.

When the submergence decreases further

to $s = 0.911$, only the breaking waves exist in the experiment. However, the present method predicts the very steep but non-breaking wave as shown in Fig.18. In the implementation of the kinematic free surface condition, the wave height is assumed to be expressed by the single-valued function of x. Thus, breaking waves or overturning waves cannot be simulated by the present scheme. However, more flexible treatment of the free surface movement such as the Lagrangian method can be adopted without difficulties. The nature of unstructured grid methods enables the spatial discretization in such a highly deformed region. This improvement together with development of the time-accurate scheme makes it possible to simulate transient breaking or overturning waves in the near future.

CONCLUSIONS

In the present study, a finite-volume method with an unstructured grid method which has been originally developed for transonic flow computations is successfully applied to incompressible flows with a free surface. The computed results for a submerged hydrofoil show good agreement with the experimental data.

Further extensions of this method are the inclusion of viscous effects, transient flow computation using time-accurate scheme, breaking or overturning waves simulation and so on. Also, the three-dimensional version of the present method, already exists for transonic flow computations[16], must be a useful tool for ship hydrodynamics or marine engineering.

ACKNOWLEDGEMENT

This work was done while the first author stayed at Princeton University as a visiting research fellow. He is grateful to Ship Research Institute, Japan and to the Science and Technology Agency of Japanese Government for the support during his stay at Princeton.

REFERENCES

[1] Kodama, Y.:"Grid Generation and Flow Computation for Practical Ship Hull Forms and Propellers Using the Geometrical Method and the IAF Scheme.", Proc. 5th Intern. Conf. Numerical Ship Hydrodynamics , Hiroshima, (1989).

[2] Chen, H.C.:"Solution of Reynolds-Averaged Navier-Stokes Equations for Three-Dimensional Incompressible Flows.", J. Comput. Phys., Vol.88, (1990).

[3] Dawson, C.W.:"A Practical Computer Method for Solving Ship Wave Problems.", Proc. 2nd Intern. Conf. Numerical Ship Hydrodynamics, Berkeley, (1977).

[4] Hino, T.:"Computation of a Free Surface Flow around an Advancing Ship by the Navier-Stokes Equations.", Proc. 5th Intern. Conf. Numerical Ship Hydrodynamics, Hiroshima, (1989).

[5] Miyata, H. et al.:"Difference Solution of a Viscous Flow with Free-Surface Wave about an Advancing Ship.", J. Comput. Phys., Vol.72, (1987).

[6] Tahara, Y. et al.:"An Interactive Approach for Calculating Ship Boundary Layers and Wakes for Nonzero Froude Number.", J. Comput. Phys., Vol.98, (1992).

[7] Jameson, A. et al.:"Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh.", AIAA J., Vol.24, No.4, (1986).

[8] Dreyer, J.D.:"Finite Volume Solutions to the Steady Incompressible Euler Equations on Unstructured Triangular Meshes.", MSE Thesis, Dept. Mechanical and Aerospace Engineering, Princeton University, (1990).

[9] Harlow F.H. et al.:"Numerical Calculation of Time-Dependent Viscous Flow of Fluid with Free Surface.", Physics of Fluids, Vol.8, (1965).

[10] Chorin, A,J.:"A Numerical Method for Solving Incompressible Viscous Flow Problems.", J. Comput. Phys., Vol.2, (1967).

[11] Jameson, A:"Computaional Transonics", Comm. Pure Appl. Math., Vol.XLI, (1988).

[12] Romante, J.E.:"Absorbing Boundary Conditions for Free Surface Waves.", J. Comput. Phys., Vol.99, (1992).

[13] Peraire, J. et al.:"Adaptive Remeshing for Compressible Flow Computations.", J. Comput. Phys., Vol.72, (1987).

[14] Mavriplis, D.:"Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes", PhD Thesis, Dept. Mechanical and Aerospace Engineering, Princeton University, (1987).

[15] Duncan, J.H.:"The Breaking and Non-Breaking Wave Resistance of a Two-Dimensional Hydrofoil.", J. Fluid Mech., Vol.126, (1983).

[16] Jameson, A. et al.:"Calculation of Inviscid Transonic Flow over a Complete Aircraft.", AIAA paper 86-0103, (1986).

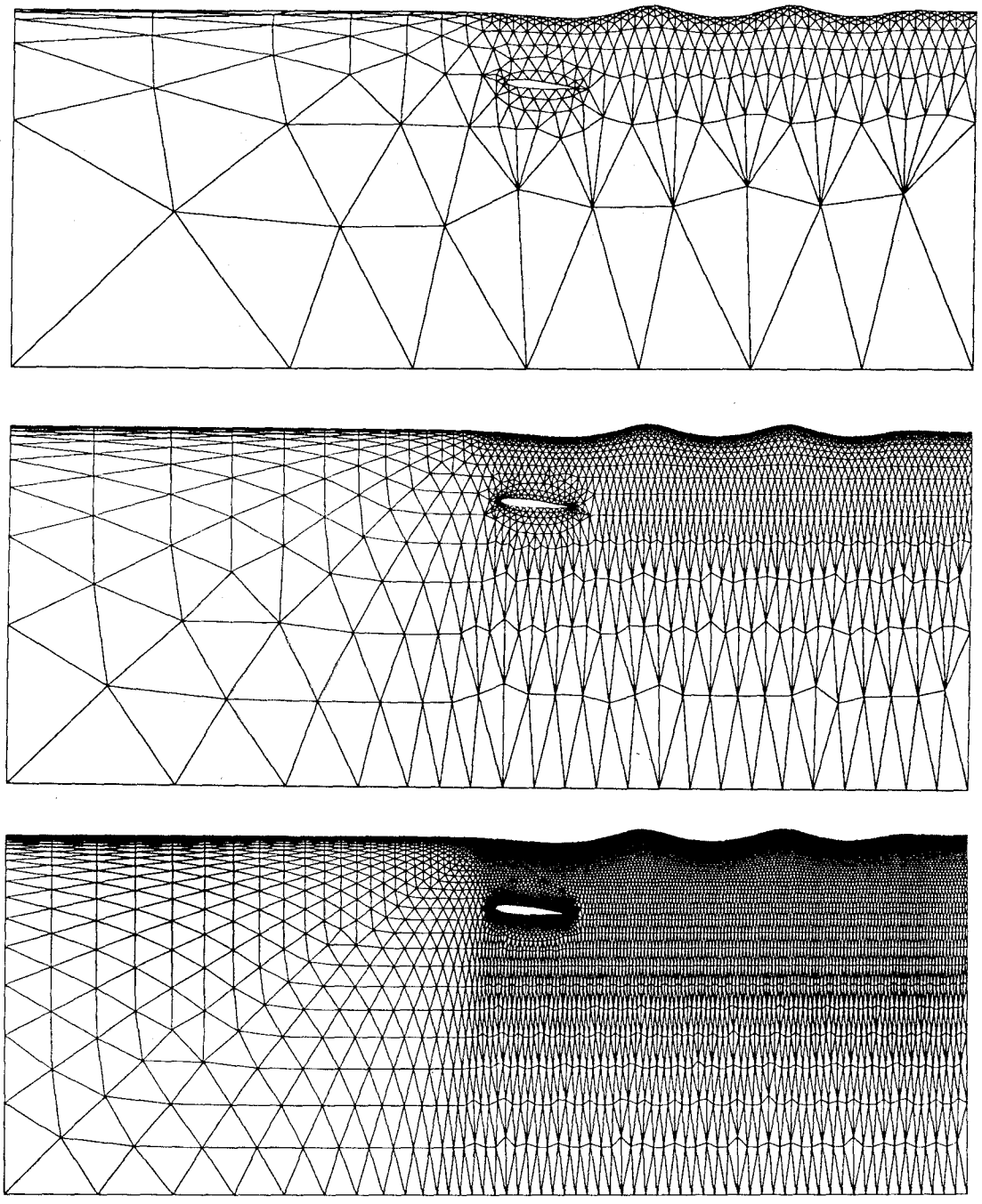

Figure 8: Sequence of multigrids around NACA0012.

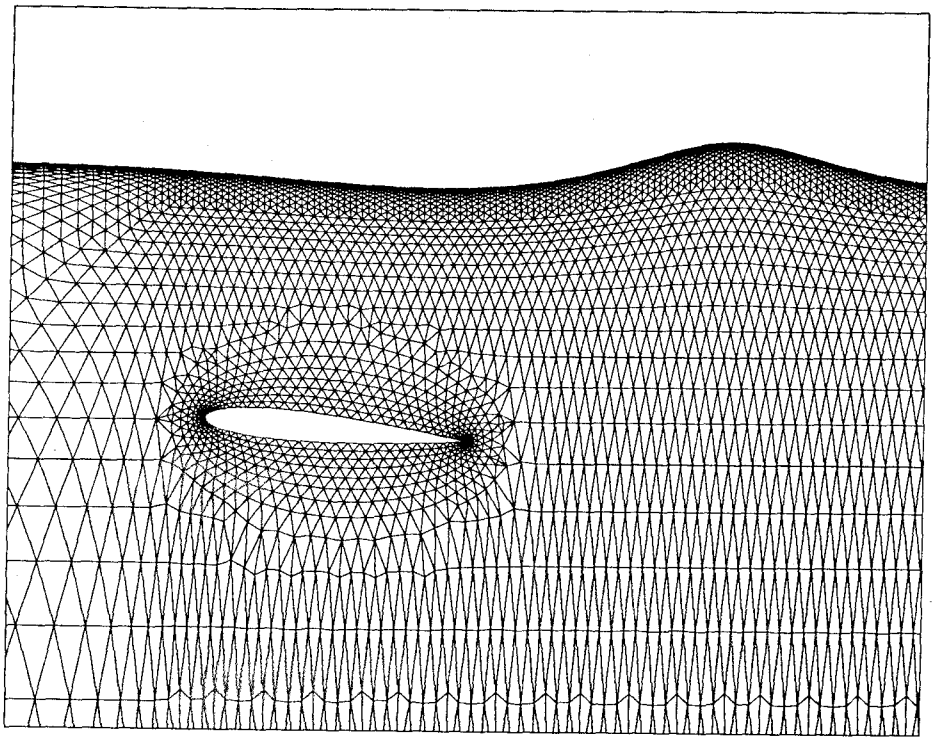

Figure 9: Magnified view of the finest grid around NACA0012.

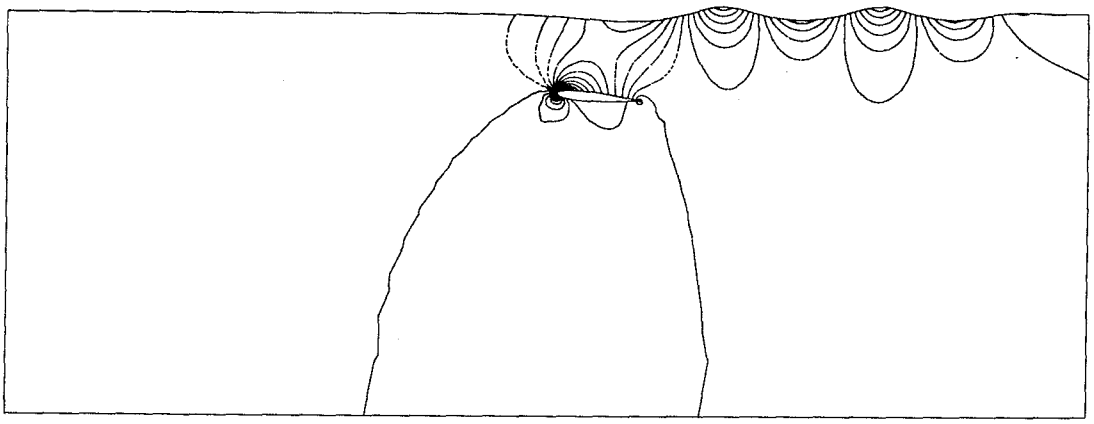

Figure 10: Computed pressure distribution with the multigrid case.

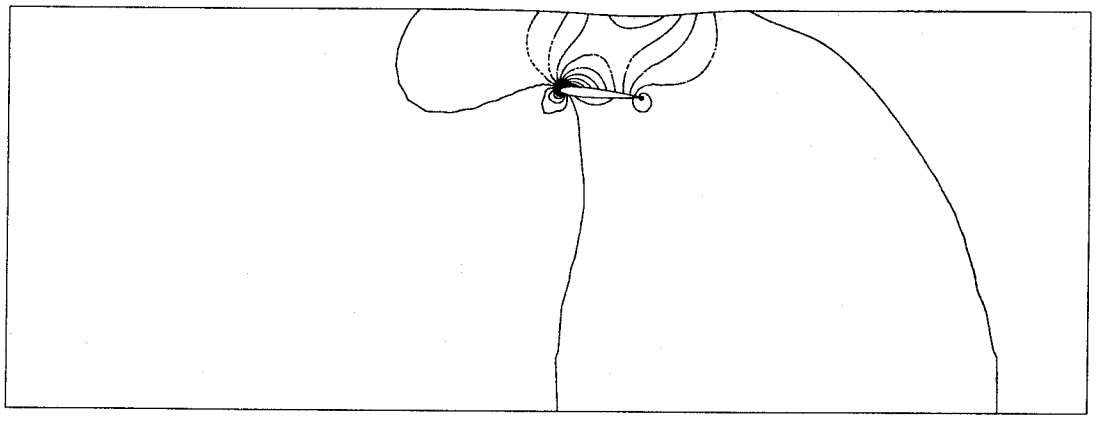

Figure 11: Computed pressure distribution with the single grid case.

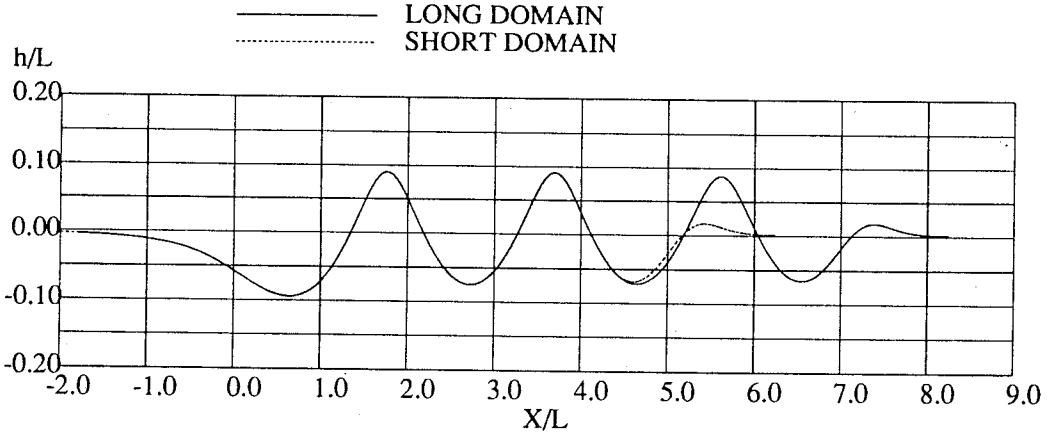

Figure 12: Comparison of wave profiles with the long and the short domains.

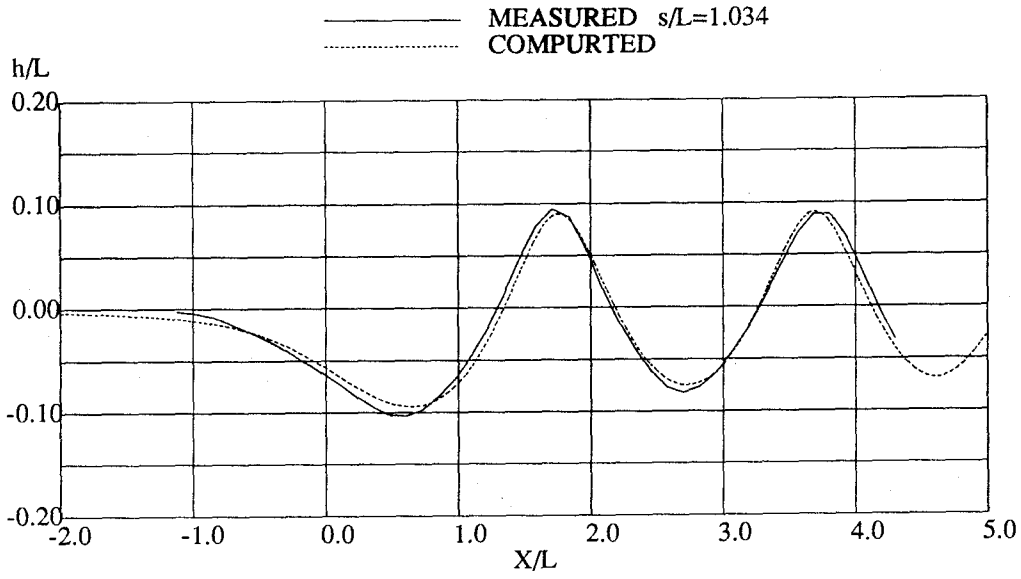

Figure 13: Comparison of wave profiles at $s = 1.034$

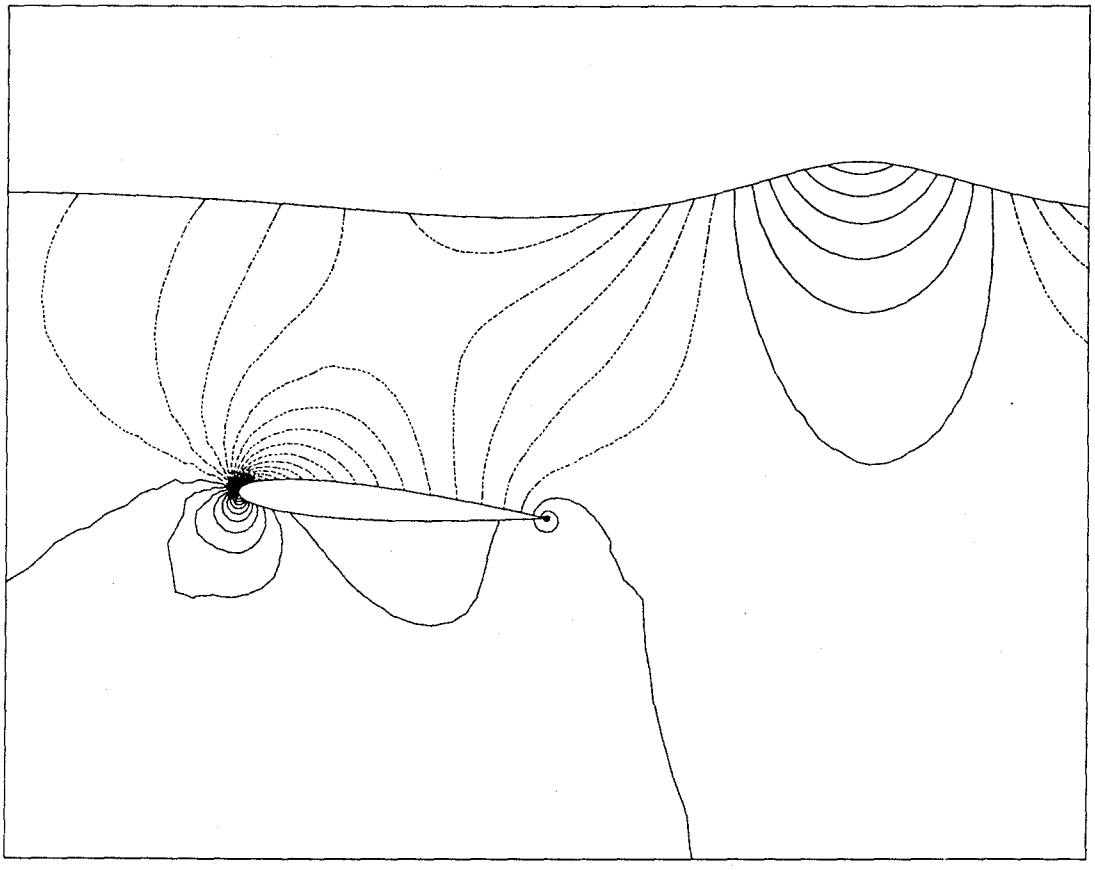

Figure 14: Computed pressure distribution.

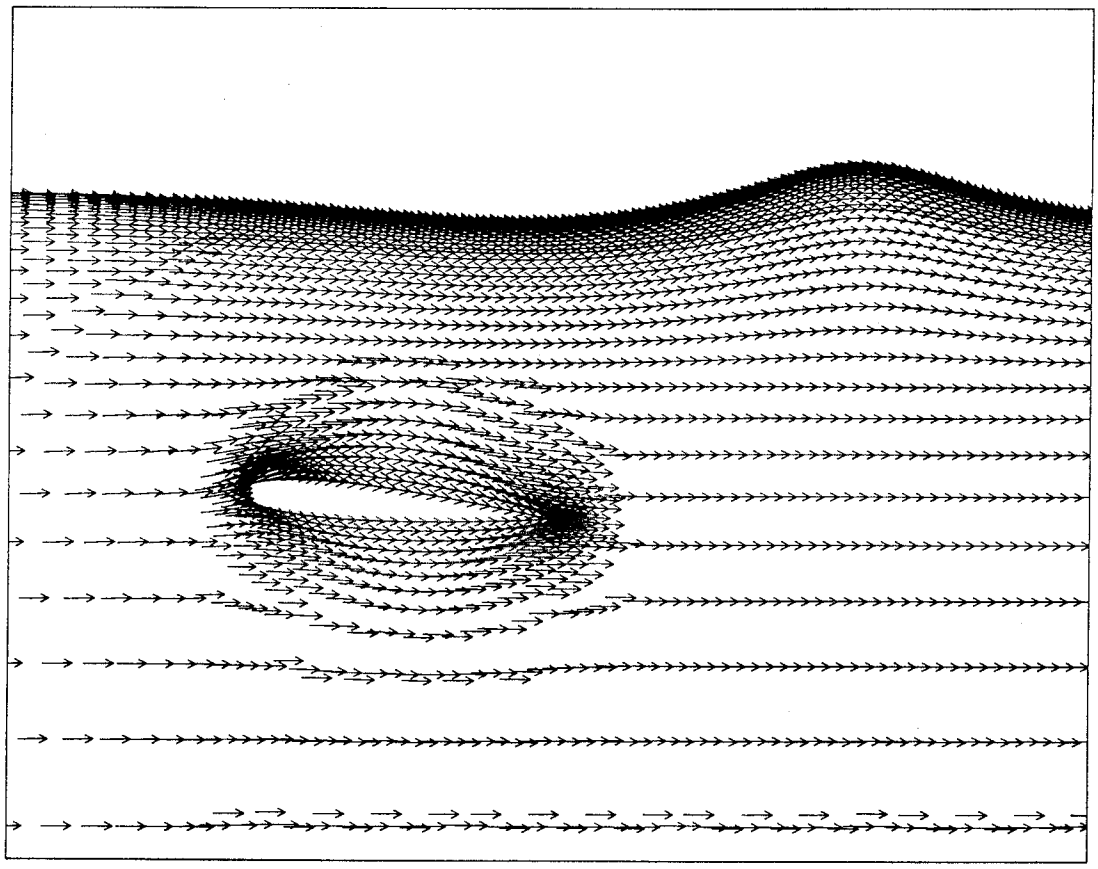

Figure 15: Computed velocity vectors.

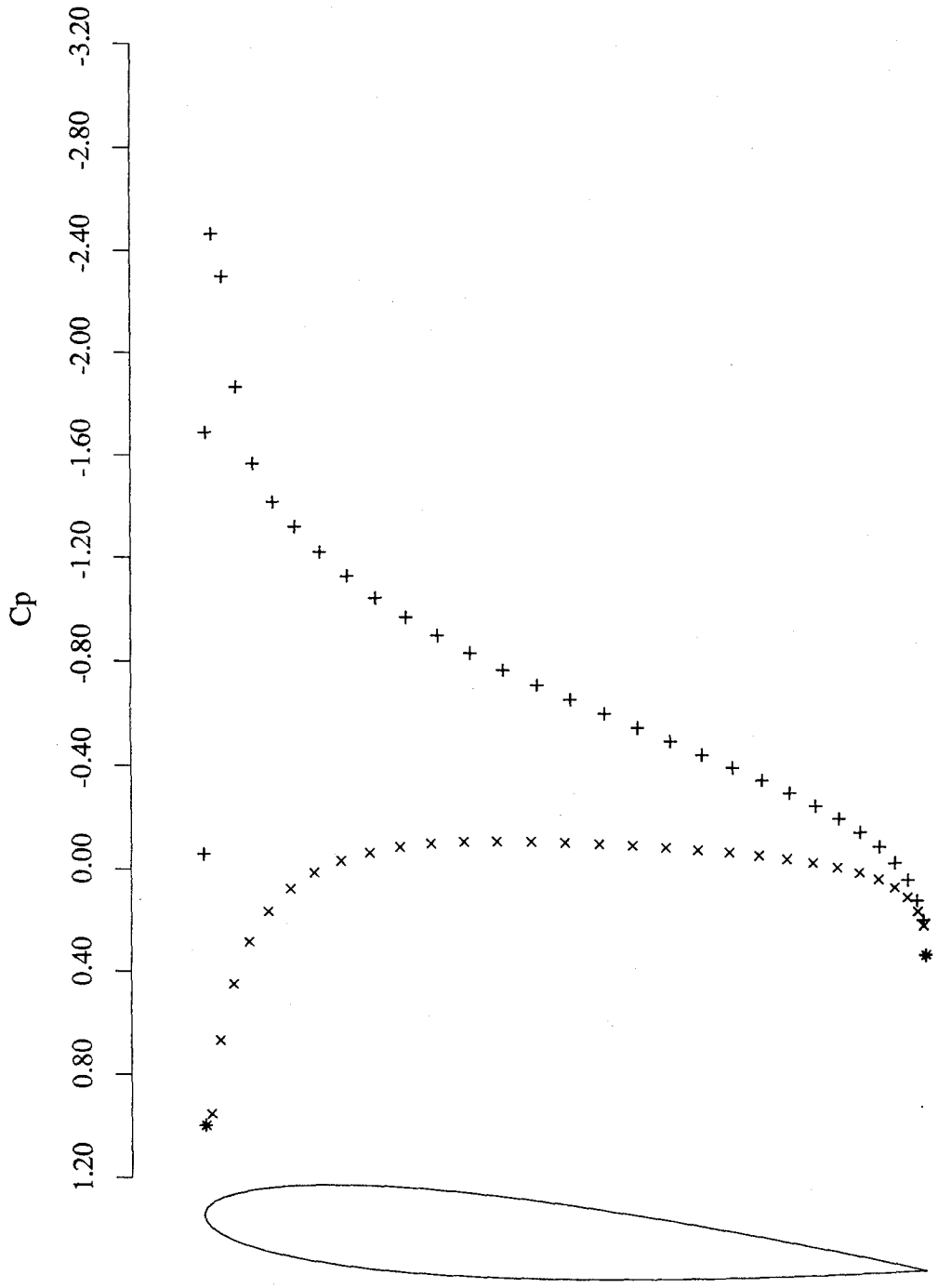

Figure 16: Computed Cp distribution.

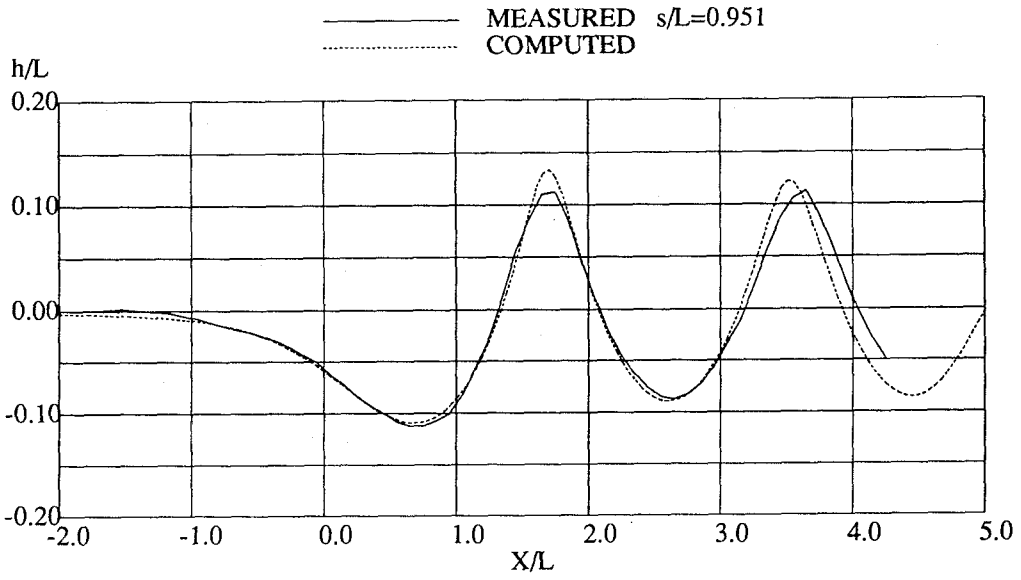

Figure 17: Comparison of wave profiles at $s = 0.951$

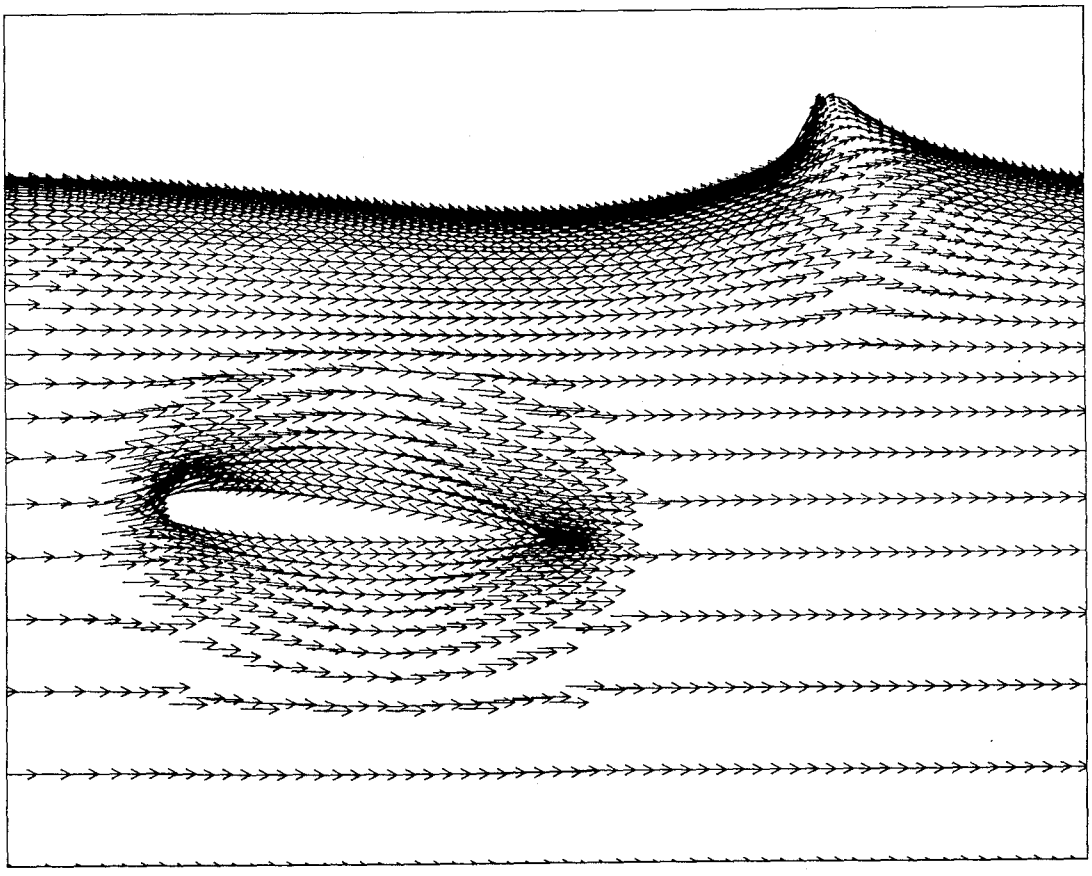

Figure 18: Computed wave profile and velocity distributions at $s = 0.911$