INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 1371–1389 (1997)

PARALLEL FINITE ELEMENT METHODS FOR LARGE-SCALE COMPUTATION OF STORM SURGES AND TIDAL FLOWS

KAZUO KASHIYAMA, KATSUYA SAITOH,[1] MAREK BEHR[2] AND TAYFUN E. TEZDUYAR[2]

[1]*Department of Civil Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112, Japan*
[2]*AEM/AHPCRC, University of Minnesota, 1100 Washington Avenue South, Minneapolis, MN 55415, U.S.A.*

SUMMARY

Massively parallel finite element methods for large-scale computation of storm surges and tidal flows are discussed here. The finite element computations, carried out using unstructured grids, are based on a three-step explicit formulation and on an implicit space–time formulation. Parallel implementations of these unstructured grid-based formulations are carried out on the Fujitsu Highly Parallel Computer AP1000 and on the Thinking Machines CM-5. Simulations of the storm surge accompanying the Ise-Bay typhoon in 1959 and of the tidal flow in Tokyo Bay serve as numerical examples. The impact of parallelization on this type of simulation is also investigated. The present methods are shown to be useful and powerful tools for the analysis of storm surges and tidal flows. ©1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **24**: 1371–1389, 1997

No. of Figures: 22. No. of Tables: 0. No. of References: 26.

KEY WORDS: parallel finite element method; three-step explicit formulation; implicit space-time formulation; storm surge; tidal flow

1. INTRODUCTION

Storm surges is a phenomenon in which the sea level in a near-shore rises significantly because of the passage of a typhoon or low atmospheric pressure. This can cause enormous damage in major bays and harbours. Tidal flows in ocean bays are less violent, yet their understanding is also important to the design of shoreline and offshore structure. For the study of storm surge, computations were carried out in the past by some of the present authors and also other researchers.[1,2] Tidal flow simulations were previously reported in References 3 and 4. The finite element method is a powerful tool in such simulations, since it is applicable to complicated water and land configurations and is able to represent such configurations accurately. In practical computations, especially in the case of storm surge analysis, the computational domain is large and the computations need to be carried out over long time periods. Therefore this type of problem becomes quite large-scale and it is essential to use methods which are as efficient and fast as the available hardware allows.

In recent years, massively parallel finite element computations have been successfully applied to several large-scale flow problems.[3,5] These computations demonstrated the availability of a new level of finite element capability to solve practical flow problems. With the need for a high-performance computing environment to carry out simulations for practical problems in storm surge analysis, in this

*Correspondence to: K. Kashiyama, Department of Civil Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112, Japan

CCC 0271–2091/97/121371–19 $17.50

Received 15 May 1996
Revised 12 July 1996

paper we present and employ a parallel explicit finite element method for computations based on unstructured grids. The finite element computations are based on a three-step explicit formulation[3] of the governing equations. In these computations we use the selective lumping technique for numerical stabilization. Parallel implementation of this unstructured-grid-based formulation is carried out on the Fujitsu Highly Parallel Computer AP1000. As a test problem, we carry out simulation of the storm surge accompanying the Ise-Bay typhoon in 1959. The computed results are compared with the observed results. The effect of parallelization on the efficiency of the computations is also examined.

The computation of the second class of problems, involving tidal flows, is accomplished here with a stabilized implicit finite element method based on the conservation variables. This stabilization method is based on the streamline upwind/Petrov–Galerkin (SUPG) formulation for compressible flows, which was originally introduced in Reference 6 for incompressible flows and in Reference 7 for the Euler equations of compressible flows. This methodology was later supplemented with a discontinuity-capturing term in References 8 and 9 and then extended in References 10 and 11 to the Navier–Stokes equations of compressible flows. The time-dependent governing equations are discretized using a space–time formulation developed for fixed domains in References 12 and 13 and for deforming domains in Reference 14. The present data-parallel implementation makes no assumptions about the structure of the computational grid and is written for the Thinking Machines CM-5 supercomputer. As a test problem, simulation of the tidal flow in Tokyo Bay is carried out.

2. GOVERNING EQUATIONS

The storm surge phenomena can be modelled using the shallow water equations, which are obtained from the conservation of momentum and mass, vertically integrated, assuming a hydrostatic pressure distribution:

$$\dot{u}_i + u_j u_{i,j} + g(\zeta - \zeta_0)_{,i} + \frac{(\tau_b)_i}{\rho(h+\zeta)} - \frac{(\tau_s)_i}{\rho(h+\zeta)} - \nu(u_{i,j} + u_{j,i})_{,j} = 0, \tag{1}$$

$$\dot{\zeta} + [(h+\zeta)u_i]_{,i} = 0, \tag{2}$$

where u_i is the mean horizontal velocity, ζ is the water elevation, h is the water depth, g is the gravitational acceleration, ζ_0 is the increase in water elevation corresponding to the atmospheric pressure drop, $(\tau_s)_i$ is the surface shear stress, $(\tau_b)_i$ is the bottom shear stress and ν is the eddy viscosity. The increase ζ_0 can be given by Fujita's formula[15] as

$$\zeta_0 = \frac{1}{10\rho g} \frac{\Delta p}{\sqrt{[1 + (r/r_0)^2]}}, \tag{3}$$

where Δp is the pressure drop at the centre of the typhoon, ρ is the density of fluid, r is the distance from the centre of the typhoon and r_0 is the radius of the typhoon.

The surface shear stress can be given as

$$(\tau_s)_i = \rho_a \gamma w_i \sqrt{(w_k w_k)}, \tag{4}$$

where ρ_a is the density of air, γ is the drag coefficient and w_i is the wind velocity 10 m above the water surface. The wind velocity can be evaluated using the expressions

$$w_1 = -\frac{C_1 V_g}{r} \{\sin\theta[x_1 - (x_1)_c] + \cos\theta[x_2 - (x_2)_c]\} + C_2 V_1 e^{-(r/R)\pi}, \tag{5}$$

$$w_2 = -\frac{C_1 V_g}{r} \{-\cos\theta[x_1 - (x_1)_c] + \sin\theta[x_2 - (x_2)_c]\} + C_2 V_2 e^{-(r/R)\pi}, \tag{6}$$

where V_g is the gradient wind velocity, V_1 and V_2 denote the velocity of the typhoon, $(x_1)_c$ and $(x_2)_c$ denote the position of the typhoon, θ is the gradient wind angle and R, C_1 and C_2 are constants. The gradient wind velocity is define as

$$V_g = \frac{fr}{2}\left[\left[-1+\sqrt{\left\{1+\frac{4\Delta p}{\rho_a f^2 r_0^2}\left[1+\left(\frac{r}{r_0}\right)^2\right]^{-3/2}\right\}}\right]\right], \tag{7}$$

where f is the Coriolis coefficient.

The bottom shear stress can be given as

$$(\tau_b)_i = \frac{n^2 g}{h^{1/3}} u_i \sqrt{(u_k u_k)}, \tag{8}$$

where n is the Manning coefficient.

3. VARIATIONAL FORMULATIONS

We present two finite element formulations of the shallow water equations which have been implemented on parallel architectures. The first method is a three-step explicit method for fixed domains. The second method is an implicit stabilized space–time formulation. Although the examples presented in this paper involve fixed domains only, the latter (space–time) formulation is seen as a step towards solving an important class of problems which involve deforming domains. With the two formulations included in this section addressing different classes of problems, a cost/accuracy comparison is not performed; however, it is expected that the explicit method for a given time step size will be more economical than the space–time formulation if the domain is fixed. The implicit space–time formulation, on the other hand, does not involve as much time step size restriction (due to numerical stability) as the explicit method.

3.1. Three-Step Explicit Finite Element Method

For the finite element spatial discretization of the governing equations the standard Galerkin method is used. The weak form of the governing equations can then be written as

$$\int_\Omega u_i^* \left(\dot{u}_i + u_j u_{i,j} + g(\zeta - \zeta_0)_{,i} + \frac{(\tau_b)_i}{\rho(h+\zeta)} - \frac{(\tau_s)_i}{\rho(h+\zeta)}\right) d\Omega + \int_\Omega u_{i,j}^* [\nu(u_{i,j} + u_{j,i})]\, d\Omega - \int_\Gamma u_i^* t_i\, d\Gamma = 0, \tag{9}$$

$$\int_\Omega \zeta^* \{\dot{\zeta} + [(h+\zeta)u_i]_{,i}\}\, d\Omega = 0, \tag{10}$$

where u_i^* and ζ^* denote the weighting functions and t_i represents boundary terms.

Using the three-node linear triangular elements for the spatial discretization, the following finite element equations can be obtained:

$$M_{\alpha\beta}\dot{u}_{\beta i} + K_{\alpha\beta\gamma j} u_{\beta j} u_{\gamma i} + H_{\alpha\beta i}(\zeta_\beta - \zeta_{0\beta}) + T_{\alpha\beta}\left(\frac{(\tau_b)_i}{\rho(h+\zeta)}\right)_\beta - T_{\alpha\beta}\left(\frac{(\tau_s)_i}{\rho(h+\zeta)}\right)_\beta + S_{\alpha i\beta j} u_{\beta j} = 0, \tag{11}$$

$$M_{\alpha\beta}\dot{\zeta}_\beta + B_{\alpha\beta i\gamma} u_{\beta i}(h_\gamma + \zeta_\gamma) + C_{\alpha\beta\gamma i} u_{\beta i}(h_\gamma + \zeta_\gamma) = 0. \tag{12}$$

The coefficient matrix can be expressed as

$$M_{\alpha\beta}=\int_\Omega \Phi_\alpha\Phi_\beta \mathrm{d}\Omega, \qquad K_{\alpha\beta\gamma j}=\int_\Omega \Phi_\alpha\Phi_\beta\Phi_{\gamma,j}\mathrm{d}\Omega,$$

$$H_{\alpha\beta i}=g\int_\Omega \Phi_\alpha\Phi_{\beta,i}\mathrm{d}\Omega, \qquad T_{\alpha\beta}=\int_\Omega \Phi_\alpha\Phi_\beta\mathrm{d}\Omega,$$

$$S_{\alpha i\beta j}=\nu\int_\Omega \Phi_{\alpha,i}\Phi_{\beta,j}\mathrm{d}\Omega+\nu\int_\Omega \Phi_{\alpha,k}\Phi_{\beta,k}\delta_{ij}\mathrm{d}\Omega,$$

$$B_{\alpha\beta i\gamma}=\int_\Omega \Phi_\alpha\Phi_{\beta,i}\Phi_\gamma\mathrm{d}\Omega, \qquad C_{\alpha\beta\gamma i}=\int_\Omega \Phi_\alpha\Phi_\beta\Phi_{\gamma,i}\mathrm{d}\Omega,$$

where Φ denotes the shape function. The bottom stress term is linearized and the water depth is interpolated using linear interpolation.

For discretization in time the three-step explicit time integration scheme is employed using the Taylor series expansion

$$F(t+\Delta t)=F(t)+\Delta t\frac{\partial F(t)}{\partial t}+\frac{\Delta t^2}{2}\frac{\partial^2 F(t)}{\partial t^2}+\frac{\Delta t^3}{6}\frac{\partial^3 F(t)}{\partial t^3}+O(\Delta t^4), \tag{13}$$

where F is an arbitrary function and Δt is the time increment. Using the approximate equation up to third-order accuracy, the following three-step scheme can be obtained:[16]

$$\begin{aligned} F\left(t+\frac{\Delta t}{3}\right)&=F(t)+\frac{\Delta t}{3}\frac{\partial F(t)}{\partial t},\\ F\left(t+\frac{\Delta t}{2}\right)&=F(t)+\frac{\Delta t}{2}\frac{\partial F(t+\Delta t/3)}{\partial t},\\ F(t+\Delta t)&=F(t)+\Delta t\frac{\partial F(t+\Delta t/2)}{\partial t}. \end{aligned} \tag{14}$$

Equation (14) is equivalent to equation (13) and the method is referred to as the three-step Taylor–Galerkin method. The stability limit of the method is 1·5 times larger than that of the conventional two-step scheme.[4,17] Details of this method are given in Reference 3. Applying this scheme to the finite element equations, the following discretized equations in time can be obtained:

Step 1

$$\begin{aligned} M^{\mathrm{L}}_{\alpha\beta}u^{n+1/3}_{\beta i}=M^{\mathrm{L}}_{\alpha\beta}u^n_{\beta i}-\frac{\Delta t}{3}\Bigg[&K_{\alpha\beta\gamma i}u^n_{\beta j}u^n_{\gamma i}+H_{\alpha\beta i}(\zeta^n_\beta-\zeta^n_{0\beta})+T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{b}})_i}{\rho(h+\zeta)}\right)^n_\beta\\ &-T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{s}})_i}{\rho(h+\zeta)}\right)^n_\beta+S_{\alpha i\beta j}u^n_{\beta j}\Bigg], \end{aligned} \tag{15}$$

$$M^{\mathrm{L}}_{\alpha\beta}\zeta^{n+1/3}_\beta=M^{\mathrm{S}}_{\alpha\beta}\zeta^n_\beta-\frac{\Delta t}{3}[B_{\alpha\beta i\gamma}u^n_{\beta i}(h_\gamma+\zeta^n_\gamma)+C_{\alpha\beta\gamma i}u^n_{\beta i}(h_\gamma+\zeta^n_\gamma)], \tag{16}$$

Step 2

$$M^{\mathrm{L}}_{\alpha\beta}u^{n+1/2}_{\beta i} = M^{\mathrm{L}}_{\alpha\beta}u^{n+1/3}_{\beta i} - \frac{\Delta t}{2}\left[K_{\alpha\beta\gamma i}u^{n+1/3}_{\beta j}u^{n+1/3}_{\gamma i} + H_{\alpha\beta i}(\zeta^{n+1/3}_{\beta} - \zeta^{n+1/3}_{0\beta}) + T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{b}})_i}{\rho(h+\zeta)}\right)^{n+1/3}_{\beta} - T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{s}})_i}{\rho(h+\zeta)}\right)^{n+1/3}_{\beta} + S_{\alpha i\beta j}u^{n+1/3}_{\beta j}\right], \tag{17}$$

$$M^{\mathrm{L}}_{\alpha\beta}\zeta^{n+1/2}_{\beta} = M^{\mathrm{S}}_{\alpha\beta}\zeta^{n+1/3}_{\beta} - \frac{\Delta t}{2}\left[B_{\alpha\beta i\gamma}u^{n+1/3}_{\beta i}(h_{\gamma} + \zeta^{n+1/3}_{\gamma}) + C_{\alpha\beta\gamma i}u^{n+1/3}_{\beta i}(h_{\gamma} + \zeta^{n+1/3}_{\gamma})\right], \tag{18}$$

Step 3

$$M^{\mathrm{L}}_{\alpha\beta}u^{n+1}_{\beta i} = M^{\mathrm{L}}_{\alpha\beta}u^{n+1/2}_{\beta i} - \Delta t\left[K_{\alpha\beta\gamma i}u^{n+1/2}_{\beta j}u^{n+1/2}_{\gamma i} + H_{\alpha\beta i}(\zeta^{n+1/2}_{\beta} - \zeta^{n+1/2}_{0\beta}) + T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{b}})_i}{\rho(h+\zeta)}\right)^{n+1/2}_{\beta} - T_{\alpha\beta}\left(\frac{(\tau_{\mathrm{s}})_i}{\rho(h+\zeta)}\right)^{n+1/2}_{\beta} + S_{\alpha i\beta j}u^{n+1/2}_{\beta j}\right], \tag{19}$$

$$M^{\mathrm{L}}_{\alpha\beta}\zeta^{n+1}_{\beta} = M^{\mathrm{S}}_{\alpha\beta}\zeta^{n+1/2}_{\beta} - \Delta t\left[B_{\alpha\beta i\gamma}u^{n+1/2}_{\beta i}(h_{\gamma} + \zeta^{n+1/2}_{\gamma}) + C_{\alpha\beta\gamma i}u^{n+1/2}_{\beta i}(h_{\gamma} + \zeta^{n+1/2}_{\gamma})\right], \tag{20}$$

where superscript n denotes the value computed at the nth time point and Δt is the time increment between the nth and the $(n+1)$th step. The coefficient $M^{\mathrm{L}}_{\alpha\beta}$ expresses the lumped coefficient and $M^{\mathrm{S}}_{\alpha\beta}$ is the selective lumping coefficient given by

$$M^{\mathrm{S}}_{\alpha\beta} = eM^{\mathrm{L}}_{\alpha\beta} + (1-e)M_{\alpha\beta}, \tag{21}$$

where e is the selective lumping parameter.

3.2. Space–Time Implicit Finite Element Method

In the implicit implementation a stabilized space–time finite element method is used. Using the conservative variables defined as

$$\mathbf{U} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} H \\ Hu_1 \\ Hu_2 \end{pmatrix},$$

where $H = h + \zeta$, the variational formulation of (1) and (2) is written as

$$\begin{aligned} &\int_{\Omega_n} \mathbf{U}^* \cdot \left(\dot{\mathbf{U}} + \mathbf{A}_i\frac{\partial \mathbf{U}}{\partial x_i}\right) \mathrm{d}Q + \int_{Q_n}\left(\frac{\partial \mathbf{U}^*}{\partial x_i}\right)\cdot\left(\mathbf{K}_{ij}\frac{\partial \mathbf{U}}{\partial x_j}\right)\mathrm{d}Q + \int_{\Omega_n}(\mathbf{U}^*)_n^+ \cdot [(\mathbf{U})_n^+ - (\mathbf{U})_n^-]\,\mathrm{d}\Omega \\ &\quad + \sum_{e=1}^{(n_{\mathrm{el}})_n}\int_{Q_n^e}\boldsymbol{\tau}(\mathbf{A}_k)^T\left(\frac{\partial \mathbf{U}^*}{\partial x_k}\right)\cdot\left[\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i\frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}\frac{\partial \mathbf{U}}{\partial x_j}\right)\right]\mathrm{d}Q \\ &\quad + \sum_{e=1}^{(n_{\mathrm{el}})_n}\int_{Q_n^e}\delta\left(\frac{\partial \mathbf{U}^*}{\partial x_i}\right)\cdot\left(\frac{\partial \mathbf{U}}{\partial x_i}\right)\mathrm{d}Q \\ &\quad = \int_{Q_n}\mathbf{U}^* \cdot \mathbf{R}\,\mathrm{d}Q + \int_{(P_n)_h}\mathbf{U}^*\mathbf{H}\,\mathrm{d}P. \end{aligned} \tag{22}$$

Here **U*** denotes the weighting function and the integration takes place over the space–time domain (or its subset referred to as slab) Q_n, its lateral boundary P_n and its lower spatial boundary Ω_n. The space–time terminology is explained in more detail in Reference 18. $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ are the coefficient matrices of the advective–diffusive system, defined as

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 1 & 0 \\ -U_1^2/H^2 + gH & 2U_1/H & 0 \\ -U_1U_2/H^2 & U_2/H & U_1/H \end{pmatrix}, \qquad \mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 1 \\ -U_1U_2/H^2 & U_2/H & U_1/H \\ -U_2^2/H^2 + gH & 0 & 2U_2/H \end{pmatrix},$$

$$\mathbf{K}_{11} = \begin{pmatrix} 0 & 0 & 0 \\ -2\nu U_1/H^2 & 2\nu/H & 0 \\ -\nu U_2/H^2 & 0 & \nu/H \end{pmatrix}, \qquad \mathbf{K}_{12} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\nu U_1/H^2 & \nu/H & 0 \end{pmatrix},$$

$$\mathbf{K}_{21} = \begin{pmatrix} 0 & 0 & 0 \\ -\nu U_2/H^2 & 0 & \nu/H \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{K}_{22} = \begin{pmatrix} 0 & 0 & 0 \\ -\nu U_1/H^2 & \nu/H & 0 \\ -2\nu U_2/H^2 & 0 & 2\nu/H \end{pmatrix}.$$

R denotes the right-hand-side vector

$$\mathbf{R} = \begin{pmatrix} 0 \\ -gH\partial(h+\zeta_0)/\partial x_1 - (\tau_b)_1/\rho + (\tau_s)_1/\rho \\ -gH\partial(h+\zeta_0)/\partial x_2 - (\tau_b)_2/\rho + (\tau_s)_2/\rho \end{pmatrix} \tag{24}$$

and **H** is the natural boundary condition term defined on the subset of the lateral boundary P_n. The notation $(\ldots)_n^+$ and $(\ldots)_n^-$ indicate the values of a discontinuous variable as the time t approaches the temporal slab boundary t_n from above and below respectively.

The first two left-hand-side terms and the entire right-hand side of equation (22) constitute the Galerkin form of the shallow water equations (1) and (2). The third term enforces weakly the continuity of the solution across the time levels t_n. The fourth and fifth terms are the SUPG stabilization and discontinuity-capturing terms respectively. For the derivation of the stabilization coefficients $\boldsymbol{\tau}$ and δ for multidimensional advection–diffusive systems see e.g. Reference 11. The stabilization terms are integrated over the interior of the space–time elements Q_n^e.

The variables and weighting functions are discretized using piecewise linear (in both space and time) interpolation functions spaces for all fields. The resulting non-linear equation system is solved using the Newton–Raphson algorithm, where at each Newton–Raphson step a coupled linear equation system is solved iteratively using the GMRES update technique.

4. PARALLEL IMPLEMENTATION

For the explicit algorithm a data-parallel implementation is performed on the Fujitsu AP1000, which is a distributed memory, highly parallel computer that supports the communication mechanism. Figure 1 shows the configuration of the AP1000 system. The AP1000 consists of 1024 processing elements which are called cells, a Sun workstation which is called the host and three independent networks which are called the T-net, B-net and S-net. Each cell possesses a memory of 16 MB. Using 1024 cells, the peak computational speed reaches 8·53 Gflops. The cells perform parallel computation synchronizing all cells and transferring boundary node data to neighbouring cells. The host performs institution of cells' environment, creation of task, transfer of data and observation

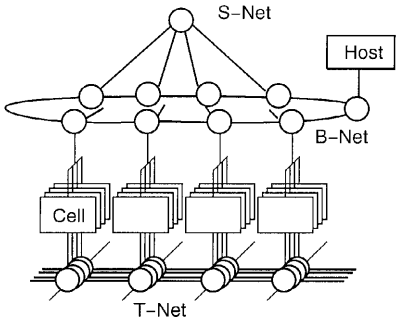

Figure 1. AP1000 system

of cells' condition. All cells are connected by the T-net (torus network) for one-to-one communication between cells. The host and cells are connected by the B-net (broadcasting network) for broadcast communication, distribution and collection of data and by the S-net (synchronization network) for barrier synchronization. The communication and synchronization mentioned above can be realized using the vendor-supplied parallel library.[19]

To minimize the amount of interprocessor communication, the automatic mesh decomposer presented by Farhat[20] is employed. For each subdomain the processor associated with that subdomain carries out computations independently, exchanging only the subdomain boundary data with the other processors.

The finite element equation can be expressed as

$$\bar{M}X = F, \tag{25}$$

where $\bar{M}$ is the lumped mass matrix, X is the unknown vector and F is the known vector. Figure 2 shows an example mesh, with the broken line denoting the boundary of a subdomain. Elements (1)–(4) belong to domain 1 (processor 1) and elements (5) and (6) belong to subdomain 2 (processor 2). The unknown values X are solved by

$$X = F/\bar{M} \tag{26}$$

No interprocessor communication is needed to compute the unknown values of a node which is located in the subdomain interior, such as node A. However, in the case of node B, which is located on the boundary of subdomains, interprocessor communication is needed and the following procedure is applied. First the following values are computed in each processor:

$$M_{B1} = M_{B(3)} + M_{B(4)}, \quad F_{B1} = F_{B(3)} + F_{B(4)} \quad (\text{processor } 1), \tag{27}$$

$$M_{B2} = M_{B(5)} + M_{B(6)}, \quad F_{B2} = F_{B(5)} + F_{B(6)} \quad (\text{processor } 2). \tag{28}$$

Next these values are gathered using the communication library, then the unknown values of node B can be obtained by

$$X_B = (F_{B1} + F_{B2})/(\bar{M}_{B1} + \bar{M}_{B2}). \tag{29}$$

Data transfer is performed at every time step (see Figure 2). As the lumped mass matrix $\bar{M}$ remains constant throughout all time step, the data transfer of that matrix is required only once.

The implicit algorithm is implemented on the Connection Machine CM-5. Similarly to the Fujitsu AP1000, the CM-5 is also a distributed memory, parallel machine, with a single partition size of up to

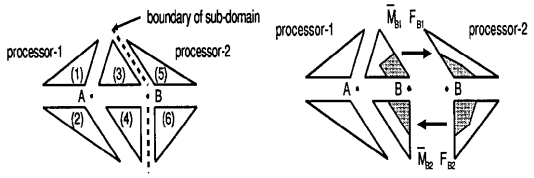

Figure 2. Parallel implementation

512 processing elements (PEs) and a Sun multiprocessor host machine. The PEs are interconnected through fat-tree data, control and diagnostic networks. Each PE manages 32 MB of memory and has a peak processing speed of 128 Mflops, for a total peak of over 65 Gflops. As on the AP1000, highly optimized communication utilities are available, grouped in the Connection Machine Scientific Software Library (CMSSL). The implementation of the implicit algorithm described in Section 3 follows closely the finite element implementation of the Navier–Stokes equations which have been described in References 21 and 22.

5. NUMERICAL EXAMPLES

As an application of the three-step explicit algorithm, simulation of the storm surge in Ise-Bay, Japan accompanying the Ise-Bay typhoon in 1959 is carried out. This typhoon occurred on 22 September 1959 and was the greatest disaster ever to hit the Ise-Bay district. Over 5000 people were killed because of this storm surge. Figure 3 shows the configuration of the domain and the path of the typhoon. Figure 4 shows the finite element discretization used. The total numbers of elements and nodes are 206,977 and 106,577 respectively. This mesh is designed to keep the element Courant number constant in the entire domain.[23,24] Figure 5 shows the water depth diagram. From Figures 4

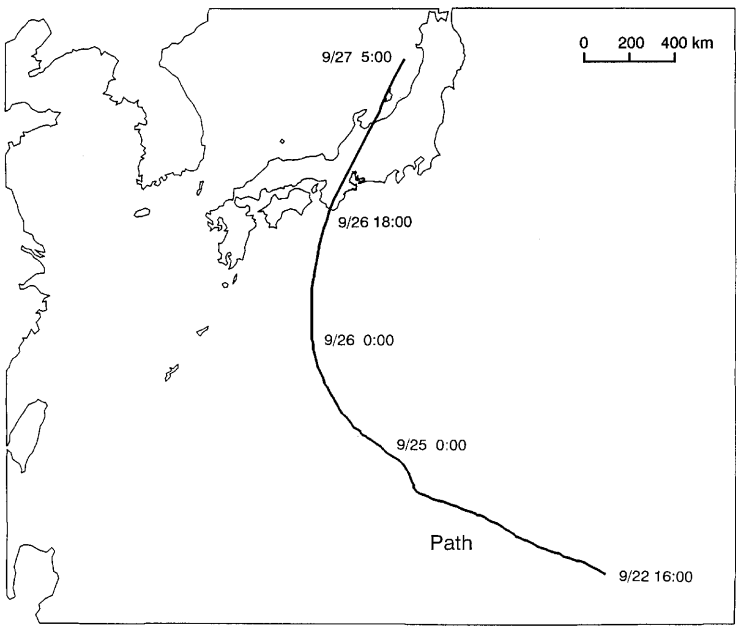

Figure 3. Computational domain and path of Ise-Bay typhoon

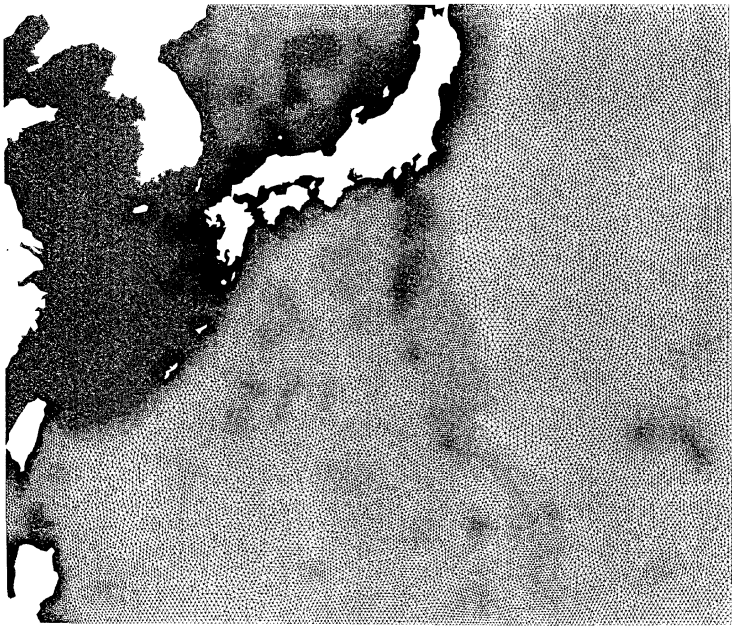

Figure 4. Finite element discretization

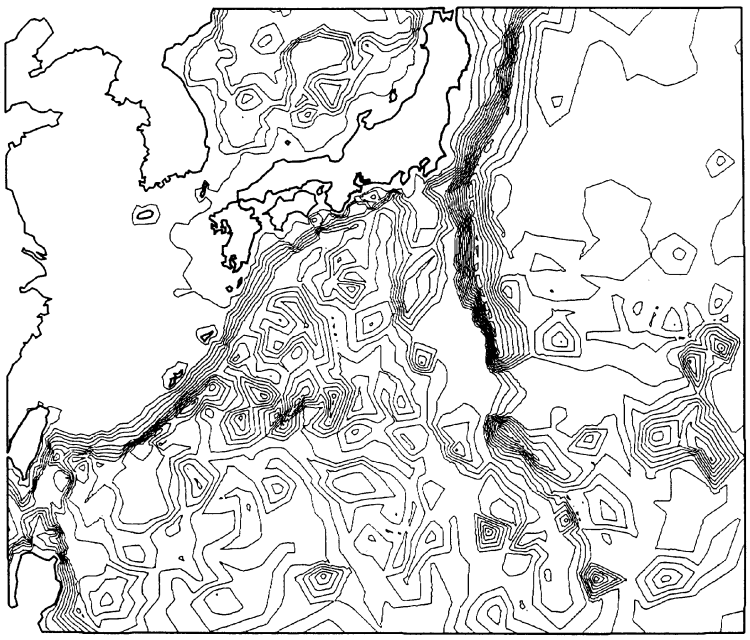

Figure 5. Water depth diagram (contours are evenly spaced at 500 m intervals)

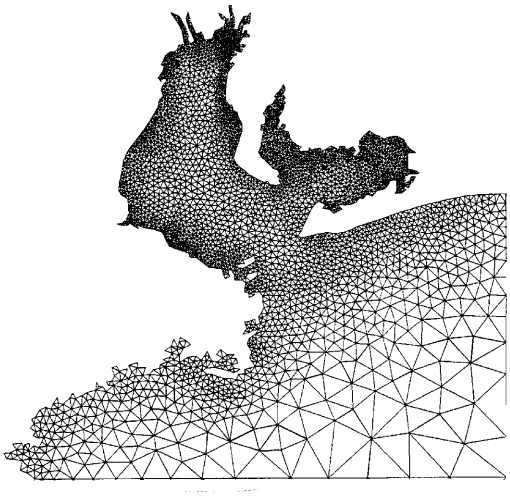

Figure 6. Finite element discretization around Ise-Bay

and 5 it can be seen that an appropriate mesh in accordance with the variation in water depth is realized. Figures 6 and 7 show the finite element discretization and water depth diagram around Ise-Bay respectively. A fine mesh which represents the geometry accurately is employed. Figure 8 shows the mesh partitioning for 512 processors. The typhoon data such as its position, speed and power are given at 1 h intervals. Using these data, the wind velocity can be computed at every time step. Linear interpolation is used for the data interpolation. For the boundary condition the no-slip boundary condition is applied to the coastline and the open-boundary condition is applied to the open boundary. For the numerical condition the following data are used: $n = 0{\cdot}3, A_1 = 10\ \mathrm{m}^2\ \mathrm{s}^{-1}, C_1 = C_2 = 0{\cdot}6, R = 500\ \mathrm{km}, r_0 = 60\ \mathrm{km}$. The selective lumping parameter and the time increment are assumed to be 0·9 and 6 s respectively. Figure 9 shows the path of the typhoons; the numerals denote the time and position of the typhoon. Figure 10 shows the computed water elevation

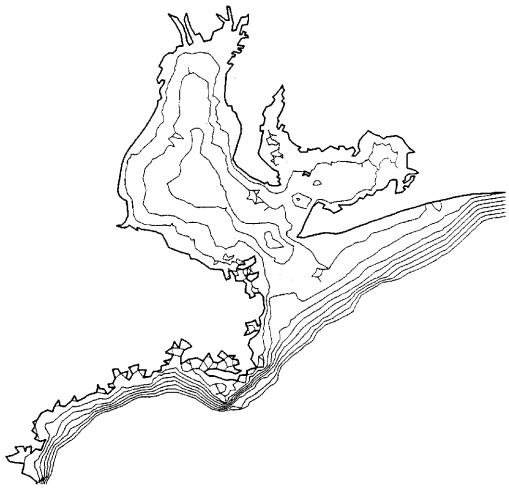

Figure 7. Water depth diagram around Ise-Bay (contours are evenly spaced at 10 m intervals)

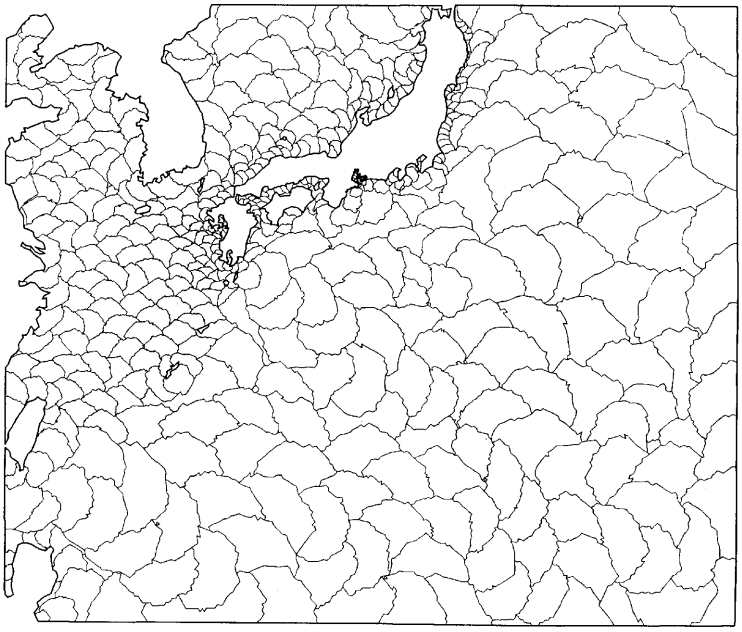

Figure 8. Mesh partitioning for 512 processors

at times 17:00 and 24:00. Figure 11 shows the computed water elevation at 1 h intervals. It can be seen that the water elevation varies according to the movement of the typhoon. Figure 12 shows the computed current velocity at time 22:00 and the complicated flow pattern. Figure 13 shows the comparison of water elevation between the computed and observed results[25] at Nagoya. It can be seen that the computed results are in good agreement with the observed results.

Figure 9. Path of typhoon

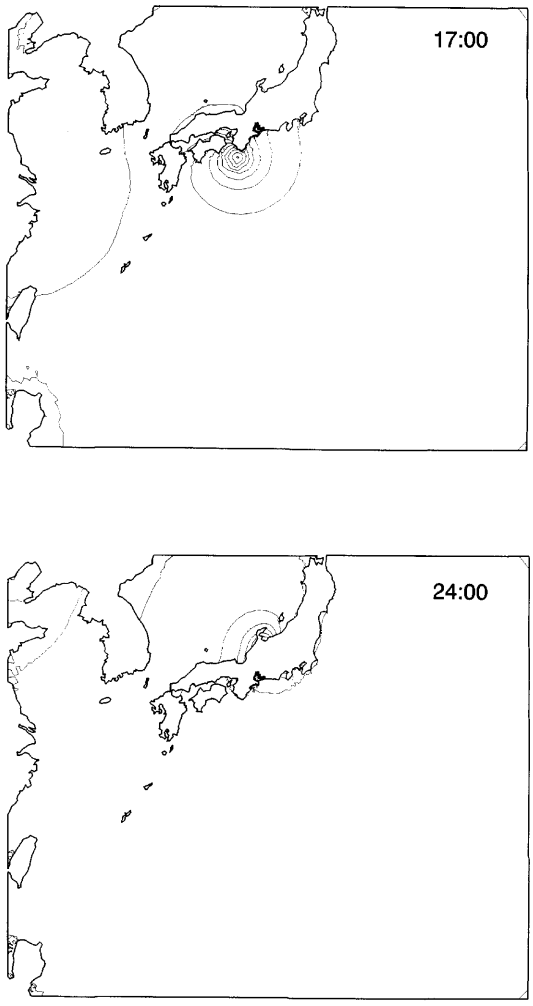

Figure 10. Computed water elevation (contours are evenly spaced at 0·1 m intervals)

In order to check the performance of the parallelization, three finite element meshes are employed: mesh L with 206,977 elements and 106,577 nodes, mesh M with 133,546 elements and 69,295 nodes and mesh S with 76,497 elements and 40,197 nodes. Figures 14 and 15 show the relation between the number of processors and the speed-up ratio and efficiency of parallelization respectively. In these figures the speed-up ratio and efficiency can be defined as

$$\text{speed-up ratio} = \frac{\text{computational time for one PE}}{\text{computational time for } N \text{ PEs}} \tag{30}$$

$$\text{efficiency} = \frac{\text{speed-up ratio}}{N}, \tag{31}$$

where N denotes the total number of processors. From these figures it can be seen that the performance is improved in accordance with an increase in the degrees of freedom and the efficiency is decreased in accordance with an increase in processors. In the case of the computation using mesh

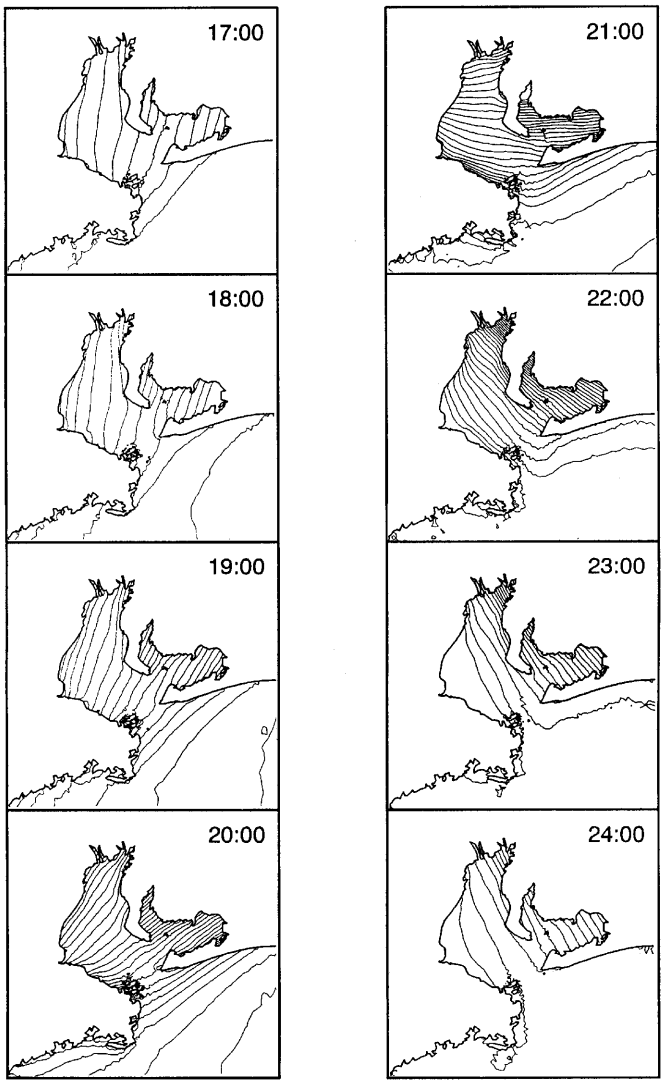

Figure 11. Computed water elevation at 1 h intervals (contours are evenly spaced at 0·1 m intervals)

L and 512 processors, it can be seen that the speed-up ratio and efficiency reach approximately 400 and 80% respectively.

As an application of the stabilized space–time formulation, the tidal flow in Tokyo Bay has been simulated. This problem was analysed earlier using the three-step explicit scheme described in Section 3.[26] Here we carry out the simulation using the implicit formulation introduced in Section 3.

The mesh used in the computation consists of 56,893 elements and 60,210 space–time nodes, as shown in Figure 16. The mesh has been decomposed into 256 subdomains (which are assigned to the individual CM-5 vector units) using a recursive spectral bisection algorithm, as shown in Figure 17. The mesh refinement is related to the water depth, shown magnified 100-fold in Figure 18. In this simulation a time step size of 60 s is chosen and the total duration is 1600 time steps, approximating

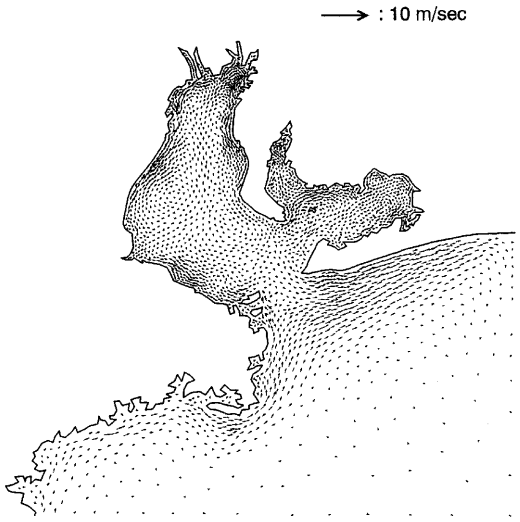

Figure 12. Computed current velocity at time 22:00

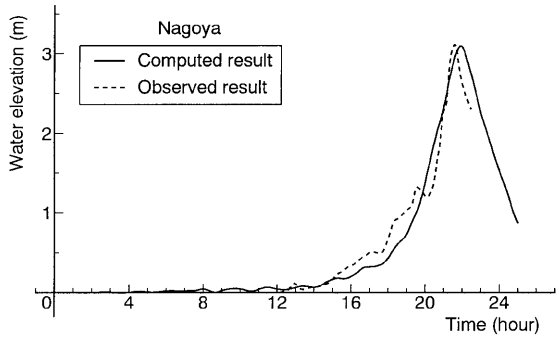

Figure 13. Comparison between computed and observed water elevation at Nagoya

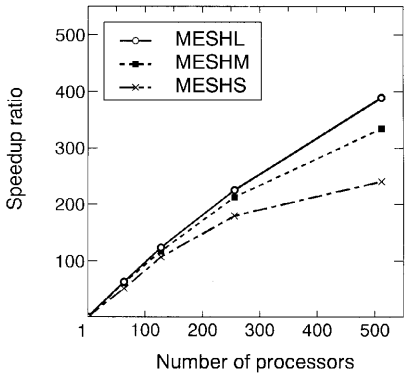

Figure 14. Comparison of speed-up ratios

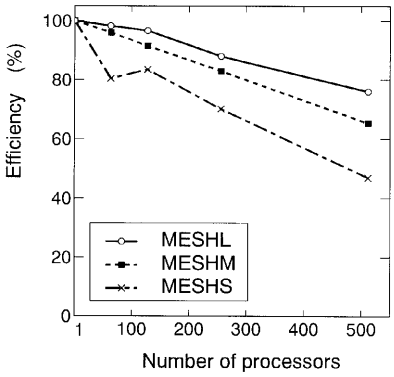

Figure 15. Comparison of efficiencies

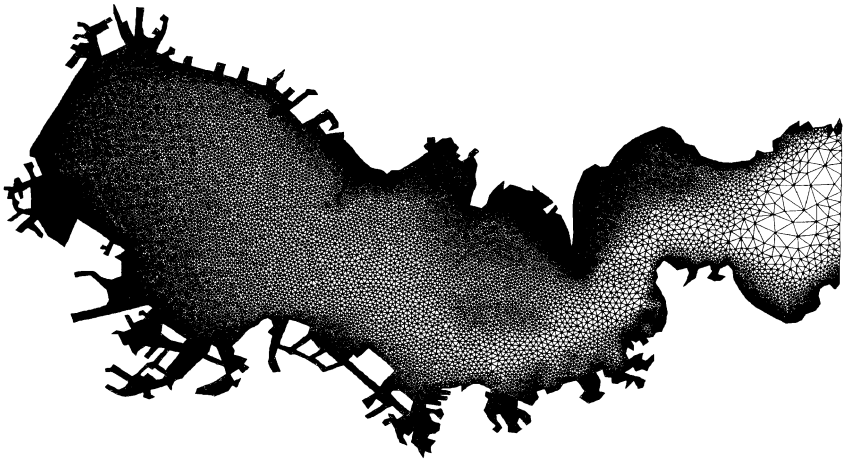

Figure 16. Finite element discretization of Tokyo Bay

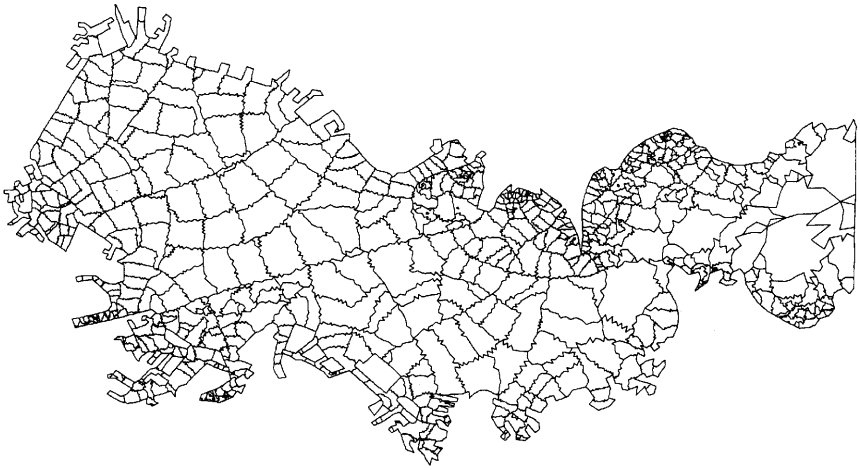

Figure 17. Mesh partitioning for 256 processors

Figure 18. Water depth view of Tokyo Bay

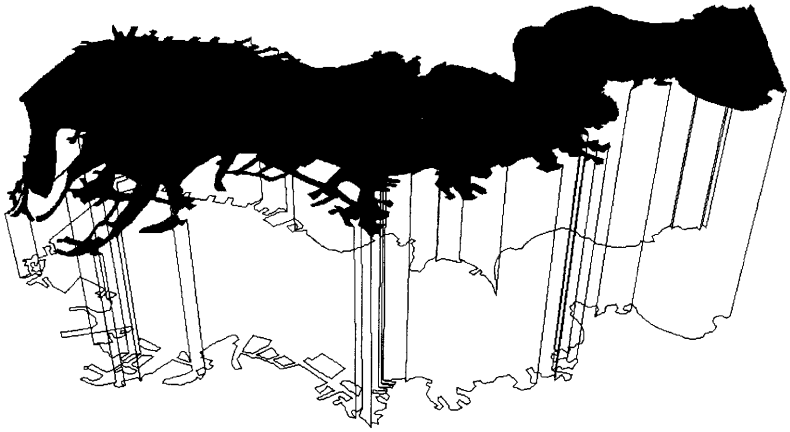

Figure 19. Computed water elevation at $t = 15{:}00$ h

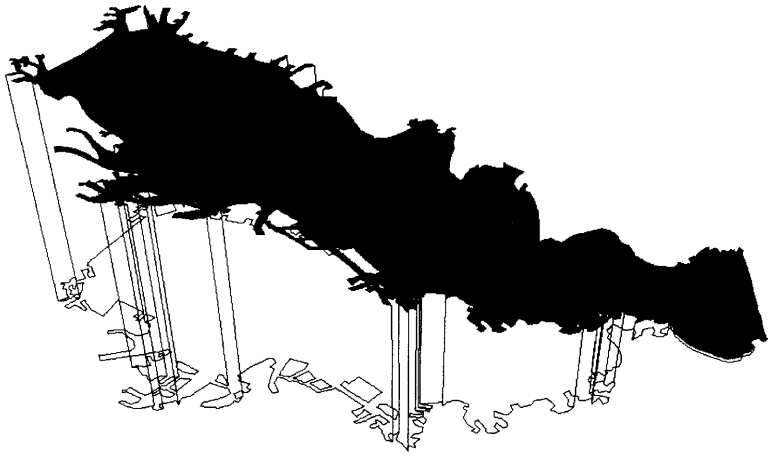

Figure 20. Computed water elevation at $t = 18{:}00$ h

Figure 21. Computed water elevation at $t = 21{:}00$ h

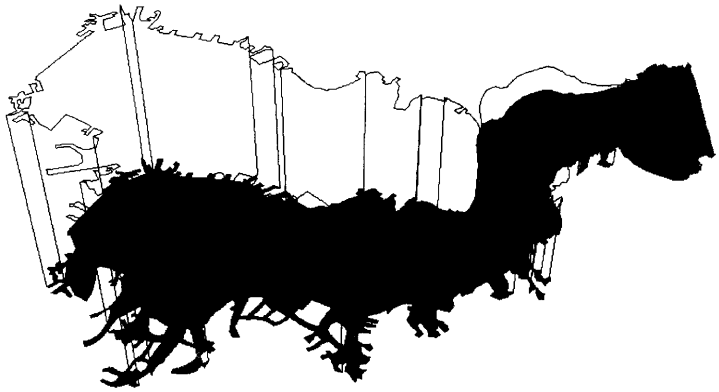

Figure 22. Computed water elevation at $t = 24{:}00$ h

one 24 h period. At the ocean boundary a diurnal tidal wave is imposed with an amplitude of 0·5 m and a period of 12 h. The following parameters are used: $n = 0{\cdot}03, A_1 = 5\ \mathrm{m}^2\ \mathrm{s}^{-1}$, $C_1 = C_2 = 0$. The storm surge term (3) is ignored in this problem. The resulting elevation is shown in Figures 19–22, magnified 50,000 times with respect to the horizontal dimensions, at times $t = 15{:}00$, 18:00, 21:00 and 24:00 h into the simulation respectively. The simulation was performed on a 64-node CM-5 with 256 vector units and took 8·5 h of computer time to complete.

6. CONCLUDING REMARKS

A three-step explicit finite element solver and an implicit stabilized space–time formulation of the shallow water equations, applicable to unstructured mesh computations of storm surges and tidal flows, have been successfully implemented on the massively parallel supercomputers AP1000 and CM-5 respectively. The explicit method has been applied to the analysis of the storm surge accompanying the Ise-Bay typhoon in 1959. The efficiency of the parallelization has been investigated and the computed results have been compared with the observed results. The performance and efficiency were observed to improve linearly in accordance with an increase in the number of degrees of freedom. The implicit method has been used to compute the tidal flow in Tokyo

Bay. From the results obtained in this paper, it can be concluded that the presented method can be successfully applied to large-scale computations of storm surges and tidal flows.

ACKNOWLEDGEMENTS

We are grateful to the Parallel Computing Research Center of Fujitsu Laboratory for providing us with access to the AP1000 resources. The last two authors were supported by ARPA and by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory co-operative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. The CRAY C90 time was provided by the Minnesota Supercomputer Institute.

REFERENCES

1. J. J. Westerink, R. A. Luettich, A. M. Baptisa, N. W. Sheffner and P. Farrar, 'Tide and storm surge predictions using finite element model', *J. Hydraul. Eng.*, **118**, 1373–1390 (1992).
2. K. Kashiyama, K. Saito and S. Yoshikawa, 'Massively parallel finite element method for large-scale computation of storm surge', *Proc. 11th Int. Conf. on Computational Methods in Water Resources*, Cancum, 1996.
3. K. Kashiyama, H. Ito, M. Behr and T. Tezduyar, 'Three-step explicit finite element computation of shallow water flows on a massively parallel computer', *Int. j. numer. methods fluids*, **21**, 885–900 (1995).
4. M. Kawahara, H. Hirano, K. Tsubota and K. Inagaki, 'Selective lumping finite element method for shallow water flow', *Int. j. numer. methods fluids*, **2**, 89–112 (1982).
5. T. E. Tezduyar, M. Behr, S. Mittal and A. A. Johnson, 'Computation of unsteady incompressible flows with the finite element methods—space–time formulations, iterative strategies and massively parallel implementations', in P. Smolinksi, W. K. Liu, G. Hulbert and K. Tamma (eds), *New Methods in Transient Analysis*, AMD Vol. 143, ASME, New York, 1992, pp. 7–24.
6. T. J. R. Hughes and A. N. Brooks, 'A multi-dimensional upwind scheme with no cross-wind diffusion', in T. J. R. Hughes (ed.), *Finite Element Methods for Convection Dominated Flows*, AMD Vol. 34, ASME, New York, 1979, pp. 19–35.
7. T. E. Tezduyar and T. J. R. Hughes, 'Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations', *AIAA Paper 83-0125*, 1983.
8. G. J. Le Beau and T. E. Tezduyar, 'Finite element computation of compressible flows with the SUPG formulation', in M. N. Dhaubhadel, M. S. Engelman and J. N. Reddy (eds), *Advances in Finite Element Analysis in Fluid Dynamics*, FED Vol. 123, ASME, New York, 1991, pp. 21–27.
9. G. J. Le Beau, S. E. Ray, S. K. Aliabadi and T. E. Tezduyar, 'SUPG finite element computation of compressible flows with the entropy and conservation variables formulations', *Comput. Methods Appl. Mech. Eng.*, **104**, 397–422 (1993).
10. S. Aliabadi, S. E. Ray and T. E. Tezduyar, 'SUPG finite element computation of compressible flows with the entropy and conservation variables formulations', *Comput. Mech.*, **11**, 300–312 (1993).
11. S. Aliabadi and T. E. Tezduyar, 'Space–time finite element computation of compresssible flows involving moving boundaries and interfaces', *Comput. Methods Appl. Mech. Eng.*, **107**, 209–224 (1993).
12. C. Johnson, U. Navert and J. Pitkäranta, 'Finite element methods for linear hyperbolic problems', *Comput. Methods Appl. Mech. Eng.*, **45**, 285–312 (1984).
13. T. J. R. Hughes and G. M. Hulbert, 'Space–time finite element methods for elastodynamics: formulations and error estimates', *Comput. Methods Appl. Mech. Eng.*, **66**, 339–363 (1988).
14. T. E. Tezduyar, M. Behr and J. Liou, 'A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary tests', *Comput. Methods Appl. Mech. Eng.*, **94**, 339–351 (1992).
15. M. Miyazaki, T. Ueno and S. Unoki, 'Theoretical investigations of typhoons surges along the Japanese coast (II)', *Oceanogr. Mag.*, **13**, 103–117 (1962).
16. C. B. Jiang, M. Kawahara, K. Hatanaka and K. Kashiyama, 'A three-step finite element method for convection–dominated incompressible flows', *Comput. Fluid Dyn. J.*, **1**, 443–462 (1993).
17. M. Kawahara and K. Kashiyama, 'Selective lumping finite element method for nearshore current', *Int. j. numer. methods fluids*, **4**, 71–97 (1984).
18. M. Behr and T. E. Tezduyar, 'Finite element solution strategies for large-scale flow simulations', *Comput. Methods Appl. Mech. Eng.*, **112**, 3–24 (1994).
19. *AP1000 Library Manual*, Fujitsu Laboratories, Tokyo, 1993.
20. C. Farhat, 'A simple and efficient automatic FEM domain decomposer', *Comput. Struct.*, **28**, 576–602 (1988).

21. M. Behr, A. Johnson, J. Kennedy, S. Mittal and T. E. Tezduyar, 'Computation of incompressible flows with implicit finite element implementation on the Connection Machine', *Comput. Methods Appl. Mech. Eng.*, **108**, 99–118 (1993).
22. J. G. Kennedy, M. Behr, V. Kalro and T. E. Tezduyar, 'Implementation of implicit finite element methods for incompressible flows on the CM-5', *Comput. Methods Appl. Mech. Eng.*, **119**, 95–111 (1994).
23. K. Kashiyama and T. Okada, 'Automatic mesh generation method for shallow water flow analysis', *Int. j. numer. methods fluids*, **15**, 1037–1057 (1992).
24. K. Kashiyama and M. Sakuraba, 'Adaptive boundary-type finite element method for wave diffraction–refraction in harbors', *Comput. Methods Appl. Mech. Eng.*, **112**, 185–197 (1994).
25. 'Reports on Ise-Bay typhoon', *Tech. Rep. 7*, Meteorological Agency, 1961 (in Japanese).
26. K. Kashiyama, H. Ito, M. Behr and T. E. Tezduyar, 'Massively parallel finite element strategies for large-scale computation of shallow water flows and contaminant transport', *Extended Abstr. Second Japan–U.S. Symp. on Finite Element Methods in Large-Scale Computational Fluid Dynamics*, Tokyo, 1994.

