ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2015, Vol. 55, No. 1, pp. 109–116. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © A.V. Kofanov, V.D. Liseikin, A.D. Rychkov, 2015, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2015, Vol. 55, No. 1, pp. 113–120.

Application of Coordinate Transformations in Numerical Simulation of Tsunami Runup by the Large Particle Method

A. V. Kofanov[a], V. D. Liseikin[a, b], and A. D. Rychkov[a]

[a] *Institute of Computational Technologies, Siberian Branch, Russian Academy of Sciences, pr. Akademika Lavrent'eva 6, Novosibirsk, 630090 Russia*

[b] *Novosibirsk State University, ul. Pirogova 2, Novosibirsk, 630090 Russia*

e-mail: avkof87@gmail.com, lvd@ict.nsc.ru, rych@ict.nsc.ru

Received September 30, 2013; in final form, March 31, 2014

Abstract—A numerical algorithm for computing the runup of a solitary tsunami wave in the case of complex shoreline topography is proposed. The algorithm involves the construction of coordinate mappings that transform a uniform rectangular grid over a reference computational domain into a grid over a physical domain with mesh refinement near the shoreline. The application of such coordinate mappings makes it possible to substantially reduce the number of grid points and save computation time. The mathematical model is based on the shallow water equations, and the problem is solved using the large particle method. An actual example is used to illustrate the computation of a curvilinear grid and the inundation area configuration.

DOI: 10.1134/S0965542515010145

Keywords: tsunami, shallow water equations, particle-in-cell method, coordinate transformations, adaptive numerical grids, inverted Beltrami and diffusion equations.

1. INTRODUCTION

The Japanese term "tsunami" means a large wave coming onshore. Such waves can be caused by underwater earthquakes, underwater volcano eruptions, or underwater landslides. Natural hazards of this type frequently lead to great damage and casualties. In the development of tools for predicting the consequences of such phenomena, a key role is played by the numerical simulation of tsunami generation, propagation, and runup. Below, this problem is addressed within the framework of the shallow water model, which adequately describes processes of this type. The main goal of the simulation is to determine the area of inundation caused by a solitary tsunami wave. Problems of this class are concerned with free-surface wave hydrodynamics, for which fairly efficient numerical methods have been developed. Primarily, these are finite-difference methods, including those based on fractional step techniques; finite element methods [1, 2]; nonconservative methods of characteristics [3]; and finite-volume methods [4, 5]. However, in the case of complex irregular bed topographies typical of actual coastal regions, most of these approaches become ineffective and special numerical algorithms are required like those proposed in [4] for the case of a lacking dry bed. Recently, much attention has been given to the design of meshless methods similar to smoothed particle hydrodynamics (SPH) [6], which is a further development of the particle-in-cell (PIC) method. In contrast to the PIC method, there is no Euler stage in SPH. As a result, the method becomes more robust and can be applied to flow computation in complex-shaped domains. On the other hand, complicated smoothing procedures are required for describing the interaction between particles; as a result, the method fails to be fully conservative. In [7] an original approach combining a finite-difference method and SPH was proposed for solving the shallow water equations in reservoirs with complex irregular bottom topography. In this paper, the runup of a tsunami wave is modeled using the well-known large particle method [8]. As in SPH, it makes use of particles in the form of elementary water columns, which are referred to hereafter as large particles. The large particle method is a development of Harlow's PIC method. It is widely used to solve evolutionary systems of differential equations associated with continuum flows and is based on the splitting of the original system of equations into physical processes. The solution process consists of two stages, Euler and Lagrangian. The Euler stage involves the evolution of flow variables concerning a cell as a whole, while the fluid is assumed to be instantly frozen. At the Lagrangian stage, the mass and momentum fluxes through the cell boundaries are determined assuming that they are caused only by the normal velocity to the boundary. This substantially simplifies the procedure for taking into account the interaction between particles, simultaneously ensuring sufficiently accurate runup (run-

down) simulation and rigorously satisfied conservation laws. Note that the number of particles required for producing SPH solutions in two-dimensional problems with the necessary accuracy is approximately 100000 (see [9]), which is comparable to the number of grid nodes in the large particle method, while the software implementation of the latter is much simpler and the operations required for computing the motion of a single large particle are roughly two orders of magnitude fewer than in SPH. Moreover, the difference scheme based on the given method is well balanced and ensures that the hydrostatic equilibrium condition holds in still water.

The effectiveness and efficiency of numerical simulation also depend considerably on the introduction of new coordinate transformations for the generation of numerical grids adaptable to fast variations in physical characteristics (the velocity of the medium), as well as to the bottom topography and the shoreline. The adaptation technique (mesh refinement in the required zones) makes it possible to considerably reduce the number of grid cells, while ensuring acceptable numerical accuracy. The construction of a coordinate transformation or a numerical grid must be automated to ensure operational predictions of possible effects of tsunami waves on the coastal zone. In this paper, the tsunami runup problem is solved by applying a modern technique for constructing coordinate transformations and numerical grids based on the solution of inverted Beltrami and diffusion equations (see [10]).

2. NUMERICAL METHOD

The shallow water equations are used to compute the runup of a solitary wave in the case of complex shoreline topography. In a Cartesian coordinate system with the $z = 0$ plane coinciding with the unperturbed water surface, the system of equations describing the motion of such a wave with allowance for depth-averaged friction is written as

$$\begin{aligned}
&\frac{\partial H}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} = 0,\\
&\frac{\partial Hu}{\partial t} + \frac{\partial Hu^2}{\partial x} + \frac{\partial Huv}{\partial y} + \frac{g}{2}\frac{\partial H^2}{\partial x} = gH\frac{\partial h}{\partial x} - uHC_R,\\
&\frac{\partial Hv}{\partial t} + \frac{\partial Huv}{\partial x} + \frac{\partial Hv^2}{\partial y} + \frac{g}{2}\frac{\partial H^2}{\partial y} = gH\frac{\partial h}{\partial y} - vHC_R,
\end{aligned} \tag{1}$$

where $H(x, y, t)$ and $h(x, y)$ are the water depth and the bottom profile, respectively, measured from the water surface; $u(x, y, t)$ and $v(x, y, t)$ are the Cartesian coordinates of the velocity vector $\mathbf{u}(x, y, t)$; C_R is the friction coefficient; and g is the acceleration due to gravity. The coefficient C_R is given by the well-known formula

$$C_R = \frac{gn^2}{H^{4/3}}|\mathbf{u}|,$$

where n is an empirical coefficient (Chézy coefficient). System (1) was solved in the rectangle $D\{0 \le x \le L_x, 0 \le y \le L_y\}$ with the bottom profile specified by bathymetry data. On the upper and lower boundaries, we used nonreflecting boundary conditions in the wave region, while, in the land region, all liquid parameters in (1) were set to zero. On the right boundary, we specified a time-varying function $H(L_x, y, t)$ taken from an analysis of an ocean wave approaching the coast.

System (1) was solved numerically by applying the large particle method [8] on an adaptive grid. The classical PIC method could not be used because the solution it produces has a statistical nature. Such a solution does not satisfy the hydrostatic equilibrium condition in still water and the resulting wave propagation is considerably distorted. The large particle method is free of this shortcoming and makes it possible to obtain a balanced difference scheme. The implementation of this method faces a difficulty associated with choosing a numerical grid. Our experience has shown that uniform rectangular grids might be used, but very fine grids are required in the entire domain to achieve acceptable accuracy and, hence, the number of grid nodes is large. Due to the constraint imposed on the time step in explicit difference schemes, this leads to considerable amounts of CPU time. Since the inundation area is of greatest interest in such problems, the number of grid nodes can be reduced using mesh refinement only near the shoreline. For example, block rectangular grids can be constructed. However, since the shoreline (which is a moving line separating the water and the land) is highly curved, the blocks have a complex irregular structure with unmatched grids on their boundaries. For this reason, we use adaptive curvilinear grids constructed by applying a time-dependent coordinate transformation

$$(x(t,\xi), y(t,\xi)) : \Xi^2 \to D, \quad \xi = (\xi^1, \xi^2) \tag{2}$$

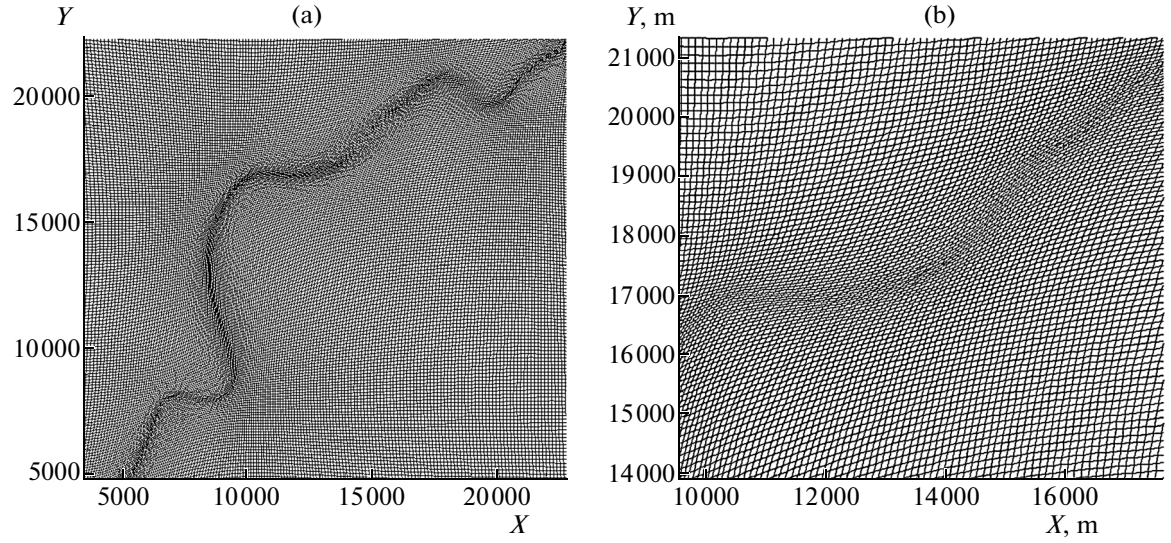

Fig. 1. Computed curvilinear grid with mesh refinement near the wave edge (left panel) and a fragment of the grid (right panel).

of the unit square computational domain $\Xi^2 = \{0 \le \xi^1, \xi^2 \le 1\}$ to the physical solution domain $D\{0 \le x \le L_x, 0 \le y \le L_y\}$. Under this transformation, the nodes of a uniform rectangular grid in Ξ^2 are mapped to nodes of a nonuniform grid in D. The coordinate transformation (2) is defined so that the cells of the resulting curvilinear grid in the physical domain are fine only near the shoreline. Since the shoreline is strongly curved (because the water coming onshore flows into troughs and rounds various elevations, which are present due to the complex land topography), the generation of a grid that is consistent with the shoreline and but is not distorted too much is a complicated problem. A certain tradeoff making it possible to construct a moving grid is that the shoreline is approximated by a sufficiently smooth curve tracing the evolution of the actual shoreline "on average." Below, we use the algorithm from [10] that generates a boundary-inconsistent grid whose cell edges may intersect the shoreline. Figure 1 shows an example of such an adaptive grid intended for the numerical solution of problem (1); the grid is finer near the shoreline. Initially, this curvilinear grid was constructed using the initial shoreline, whose position was determined by the still water condition. As a wave approached the shore, the shoreline position varied; accordingly, a new curve approximating the new shoreline on average was constructed at every time step and the grid was updated. In the case of a fixed grid, there was no updating, i.e., a grid was constructed only once.

With the use of coordinate transformation (2), the runup problem can be solved numerically on a uniform grid in Ξ^2. For this purpose, Eqs. (1) are rewritten in curvilinear coordinates t, ξ^1, and ξ^2 in conservative form with the help of a tensor identity relating the divergence of a vector in Cartesian coordinates t, $x^1, \ldots, x^n$ to that in curvilinear coordinates t, $\xi^1, \ldots, \xi^n$:

$$\frac{\partial A^0}{\partial t} + \sum_{i=1}^{n} \frac{\partial A^i}{\partial x^i} = \frac{1}{J}\left(\frac{\partial}{\partial t}(JA^0) + \sum_{j=1}^{n} \frac{\partial}{\partial \xi^j}\left[J(\bar{A}^j - A^0 \bar{w}^j) \right] \right),$$

where

$$J = \det\left\{ \frac{\partial x^i}{\partial \xi^j} \right\}, \quad \bar{A}^j = \sum_{i=1}^{n} A^i \frac{\partial \xi^j}{\partial x^i}, \quad \bar{w}^j = \sum_{i=1}^{n} \frac{\partial x^i}{\partial t} \frac{\partial \xi^j}{\partial x^i}.$$

In the variables t, ξ^1, and ξ^2, system (1) is written with the help of this tensor identity in the following conservative form:

$$\frac{\partial (JH)}{\partial t} + \frac{\partial}{\partial \xi^1}\left\{ H\left[\left(u - \frac{\partial x}{\partial t} \right) \frac{\partial y}{\partial \xi^2} - \left(v - \frac{\partial y}{\partial t} \right) \frac{\partial x}{\partial \xi^2} \right] \right\} + \frac{\partial}{\partial \xi^2}\left\{ H\left[\left(v - \frac{\partial y}{\partial t} \right) \frac{\partial x}{\partial \xi^1} - \left(u - \frac{\partial x}{\partial t} \right) \frac{\partial y}{\partial \xi^1} \right] \right\} = 0,$$

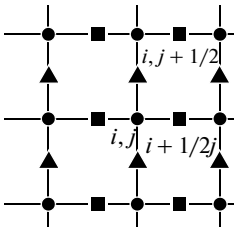

Fig. 2. Structure of the numerical grid.

$$\frac{\partial(JHu)}{\partial t}+\frac{\partial}{\partial\xi^1}\left\{Hu\left[\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi^2}-\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi^2}\right]\right\}+\frac{\partial}{\partial\xi^2}\left\{Hu\left[\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi^1}-\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi^1}\right]\right\}$$
$$+\frac{g}{2}\left[\frac{\partial H^2}{\partial\xi^1}\frac{\partial y}{\partial\xi^2}-\frac{\partial H^2}{\partial\xi^2}\frac{\partial y}{\partial\xi^1}\right]=gH\left[\frac{\partial h}{\partial\xi^1}\frac{\partial y}{\partial\xi^2}-\frac{\partial h}{\partial\xi^2}\frac{\partial y}{\partial\xi^1}\right]-JuHC_R,\tag{3}$$

$$\frac{\partial(JHv)}{\partial t}+\frac{\partial}{\partial\xi^1}\left\{Hv\left[\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi^2}-\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi^2}\right]\right\}+\frac{\partial}{\partial\xi^2}\left\{Hv\left[\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi^1}-\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi^1}\right]\right\}$$
$$+\frac{g}{2}\left[\frac{\partial H^2}{\partial\xi^2}\frac{\partial x}{\partial\xi^1}-\frac{\partial H^2}{\partial\xi^1}\frac{\partial x}{\partial\xi^2}\right]=gH\left[\frac{\partial h}{\partial\xi^2}\frac{\partial x}{\partial\xi^1}-\frac{\partial h}{\partial\xi^1}\frac{\partial x}{\partial\xi^2}\right]-JvHC_R,$$

where $J=\dfrac{\partial x}{\partial\xi^1}\dfrac{\partial y}{\partial\xi^2}-\dfrac{\partial x}{\partial\xi^2}\dfrac{\partial y}{\partial\xi^1}$.

The components $x(t, \xi)$ and $y(t, \xi)$ of coordinate transformation (2), as the water flow parameters in system (3), are found at nodes of a uniform rectangular grid in the reference domain Ξ^2. No reinterpolation of these parameters in the physical domain D is required.

The large particle method as applied to the solution of Eqs. (3) was implemented in two stages. At the first (Euler) stage, all convective terms were dropped from system (3), which was written as

$$\frac{\partial JH}{\partial t}=0,$$
$$\frac{\partial u}{\partial t}+g\left[\frac{\partial(H-h)}{\partial\xi_1}\frac{\partial y}{\partial\xi_2}-\frac{\partial(H-h)}{\partial\xi_2}\frac{\partial y}{\partial\xi_1}\right]=-uC_R,\tag{4}$$
$$\frac{\partial v}{\partial t}+g\left[\frac{\partial(H-h)}{\partial\xi_2}\frac{\partial x}{\partial\xi_1}-\frac{\partial(H-h)}{\partial\xi_1}\frac{\partial x}{\partial\xi_2}\right]=-vC_R.$$

At the second (Lagrangian) stage, we solved the advection equations

$$\frac{\partial(JH)}{\partial t}+\frac{\partial}{\partial\xi_1}\left\{H\left[\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_2}-\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_2}\right]\right\}+\frac{\partial}{\partial\xi_2}\left\{H\left[\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_1}-\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_1}\right]\right\}=0,$$
$$\frac{\partial(JHu)}{\partial t}+\frac{\partial}{\partial\xi_1}\left\{Hu\left[\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_2}-\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_2}\right]\right\}+\frac{\partial}{\partial\xi_2}\left\{Hu\left[\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_1}-\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_1}\right]\right\}=0,\tag{5}$$
$$\frac{\partial(JHv)}{\partial t}+\frac{\partial}{\partial\xi_1}\left\{Hv\left[\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_2}-\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_2}\right]\right\}+\frac{\partial}{\partial\xi_2}\left\{Hv\left[\left(v-\frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial\xi_1}-\left(u-\frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial\xi_1}\right]\right\}=0.$$

Equations (4) and (5) were solved by applying a first-order accurate explicit difference scheme. It was constructed on a rectangular staggered grid in Ξ^2, whose structure is given in Fig. 2. The grid functions $H_{i,j}$ and $h_{i,j}$ (circles) were determined at grid nodes. The velocity components u and v are denoted by squares and triangles, respectively.

Computations have shown that the numerical solution found on this grid does not exhibit spurious oscillations on the irregular bottom topography, which took place in the case of nonstaggered nodes. The difference scheme at the first stage was written as

$$J\tilde{H}_{i,j} = (JH)^n_{i,j},$$

$$\tilde{u}_{i+1/2,j} = \{u^n_{i+1/2,j} - \Delta t \cdot g[[(H-h)^n_{i+1,j} - (H-h)^n_{i,j}]/\Delta\xi_1 \cdot (y^n_{i+1/2,j+1/2} - y^n_{i+1/2,j-1/2})/\Delta\xi_2$$
$$-[(H-h)^n_{i+1/2,j+1/2} - (H-h)^n_{i+1/2,j-1/2}]/\Delta\xi_2 \cdot (y^n_{i+1,j} - y^n_{i,j})/\Delta\xi_1]\}/[1 + \Delta t(C_R)^n_{i+1/2,j}],$$

$$\tilde{v}_{i,j+1/2} = \{v^n_{i,j+1/2} - \Delta t \cdot g[[(H-h)^n_{i,j+1} - (H-h)^n_{i,j}]/\Delta\xi_2 \cdot (x^n_{i+1/2,j+1/2} - x^n_{i-1/2,j+1/2})/\Delta\xi_1$$
$$-[(H-h)^n_{i+1/2,j+1/2} - (H-h)^n_{i-1/2,j+1/2}]/\Delta\xi_1 \cdot (x^n_{i,j+1} - x^n_{i,j})/\Delta\xi_2]\}/[1 + \Delta t(C_R)^n_{i,j+1/2}].$$

At the second stage, advection equations (5) were solved using the upwind difference scheme

$$J\tilde{H}_{i,j} = (JH)^n_{i,j},$$

$$\tilde{u}_{i+1/2,j} = \{u^n_{i+1/2,j} - \Delta t \cdot g[[(H-h)^n_{i+1,j} - (H-h)^n_{i,j}]/\Delta\xi_1 \cdot (y^n_{i+1/2,j+1/2} - y^n_{i+1/2,j-1/2})/\Delta\xi_2$$
$$-[(H-h)^n_{i+1/2,j+1/2} - (H-h)^n_{i+1/2,j-1/2}]/\Delta\xi_2 \cdot (y^n_{i+1,j} - y^n_{i,j})/\Delta\xi_1]\}/[1 + \Delta t(C_R)^n_{i+1/2,j}],$$

$$\tilde{v}_{i,j+1/2} = \{v^n_{i,j+1/2} - \Delta t \cdot g[[(H-h)^n_{i,j+1} - (H-h)^n_{i,j}]/\Delta\xi_2 \cdot (x^n_{i+1/2,j+1/2} - x^n_{i-1/2,j+1/2})/\Delta\xi_1$$
$$-[(H-h)^n_{i+1/2,j+1/2} - (H-h)^n_{i-1/2,j+1/2}]/\Delta\xi_1 \cdot (x^n_{i,j+1} - x^n_{i,j})/\Delta\xi_2]\}/[1 + \Delta t(C_R)^n_{i,j+1/2}],$$

$$(JH)^{n+1}_{i,j} = (J\tilde{H})_{i,j} - \Delta t[(\tilde{q}_{i+1/2,j} - \tilde{q}_{i-1/2,j})/\Delta\xi_1 + (\tilde{q}_{i,j+1/2} - \tilde{q}_{i,j-1/2})/\Delta\xi_2],$$

$$\tilde{q}_{i+1/2,j} = \begin{Bmatrix} \tilde{H}_{i,j}U^{n+1}_{i+1/2,j}, & U^{n+1}_{i+1/2,j} > 0, \\ \tilde{H}_{i+1,j}U^{n+1}_{i+1/2,j}, & U^{n+1}_{i+1/2,j} < 0, \end{Bmatrix}, \quad \tilde{q}_{i,j+1/2} = \begin{Bmatrix} \tilde{H}_{i,j}U^{n+1}_{i,j+1/2}, & U^{n+1}_{i,j+1/2} > 0, \\ \tilde{H}_{i,j+1}U^{n+1}_{i,j+1/2}, & U^{n+1}_{i,j+1/2} < 0, \end{Bmatrix},$$

$$U^{n+1}_{i+1/2,j} = \{[\tilde{u}_{i+1/2,j} - (x^{n+1}_{i+1/2,j} - x^n_{i+1/2,j})/\Delta t] \cdot (y^{n+1}_{i+1/2,j+1/2} - y^{n+1}_{i+1/2,j-1/2})/\Delta\xi_2$$
$$-[\tilde{v}_{i+1/2,j} - (y^{n+1}_{i+1/2,j} - y^n_{i+1/2,j})/\Delta t] \cdot (x^{n+1}_{i+1,j} - x^{n+1}_{i,j})/\Delta\xi_1\},$$

$$U^{n+1}_{i,j+1/2} = \{[\tilde{u}_{i,j+1/2} - (x^{n+1}_{i,j+1/2} - x^n_{i,j+1/2})/\Delta t] \cdot (y^{n+1}_{i,j+1} - y^{n+1}_{i,j})/\Delta\xi_2$$
$$-[\tilde{v}_{i,j+1/2} - (y^{n+1}_{i,j+1/2} - y^n_{i,j+1/2})/\Delta t] \cdot (x^{n+1}_{i+1,j+1/2} - x^{n+1}_{i,j+1/2})/\Delta\xi_1\}.$$

The expressions for computing $(JHu)^{n+1}_{i+1/2,j}$ and $(JHv)^{n+1}_{i,j+1/2}$ were written in a similar manner. To ensure that the hydrostatic equilibrium conditions in still water were satisfied at a difference level, the value of the Jacobian J in system (5) at a new time level was determined from the differential identity

$$\frac{\partial J}{\partial t} + \frac{\partial}{\partial\xi_1}\left\{\left[-\frac{\partial x}{\partial t}\frac{\partial y}{\partial\xi_2} + \frac{\partial y}{\partial t}\frac{\partial x}{\partial\xi_2}\right]\right\} + \frac{\partial}{\partial\xi_2}\left\{\left[-\frac{\partial y}{\partial t}\frac{\partial x}{\partial\xi_1} + \frac{\partial x}{\partial t}\frac{\partial y}{\partial\xi_1}\right]\right\} \equiv 0$$

by applying a difference scheme similar to that described above.

In the case of a moving grid, the shoreline was again approximated by a sufficiently smooth curve as described above, the grid was updated, and its new coordinates were used to compute the node velocities $\partial x/\partial t$ and $\partial y/\partial t$.

At all grid nodes, the stability condition for difference scheme (4), (5) was given by

$$\Delta t \le \alpha \cdot \min\left\{\frac{\Delta x}{|u| + \sqrt{gH}}, \frac{\Delta y}{|v| + \sqrt{gH}}\right\},$$

where $0 < \alpha < 1$ is an empirical stability margin coefficient. It was found that $\alpha < 0.3$ ensured a stable solution in the case of nonstaggered nodes, while $\alpha = 0.5$ was sufficient on a staggered grid.

The problem was solved using two grids: fixed in the computational domain Ξ^2 and moving in the physical domain D; the latter was obtained with the help of the method described in [10, 11]. As a reference curve, we used an "averaged" shoreline as described above. The positions of points on the actual shoreline were determined at every time by analyzing whether neighboring nodes belonged to the land or water areas. The flow parameters on the shoreline were computed using one-sided differences, since all liquid parameters in the land area were set to zero.

3. CONSTRUCTION OF THE COORDINATE TRANSFORMATIONS

To solve the given problem, we constructed a discrete coordinate transformation (2) that was able to adapt to the time-varying shoreline. A method for constructing such n-dimensional transformations was described in detail in [10–13]. Below, we briefly describe it in the two-dimensional case. The two-dimensional nonstationary coordinate transformation

$$x(t,\xi):\Xi^2\to X^2,\quad \mathbf{x}(t,\xi)=(x^1(t,\xi),x^2(x,\xi)),\quad \xi=\xi^1,\xi^2,$$

where $x^1=x$ and $x^2=y$, is found at nodes of a reference grid introduced in the computational domain Ξ^2 by numerically solving inverted Beltrami or diffusion equations for a monitor metric (see [12]). More specifically, for the spherical metric tensor of the monitor metric, the inverted diffusion equations have the form

$$g_{22}\frac{\partial^2 x^k}{\partial\xi^1\partial\xi^1}-2g_{12}\frac{\partial^2 x^k}{\partial\xi^1\partial\xi^2}+g_{11}\frac{\partial^2 x^k}{\partial\xi^2\partial\xi^2}=J^2\frac{1}{Z(\mathbf{x})\partial x^k}Z(\mathbf{x}),\quad k=1,2, \tag{6}$$

where

$$g_{ij}=\frac{\partial x^1}{\partial\xi^i}\frac{\partial x^1}{\partial\xi^j}+\frac{\partial x^2}{\partial\xi^i}\frac{\partial x^2}{\partial\xi^j},\quad J=\det\left(\frac{\partial x^i}{\partial\xi^j}\right),\quad i,j=1,2,$$

$Z(\mathbf{x})$ is a positive monitor function.

The numerical solution of Eqs. (6) on a uniform grid in Ξ^2 produces a grid with mesh refinement in zones of D where $Z(x)$ is small and with mesh coarsening in zones where $Z(x)$ is large (see [10, 11, 13]). Numerical algorithms for adaptive grid generation based on the solution of these equations can be found in [11, 13]. Inverted diffusion equations were used for adaptive mesh generation in numerical computations concerning nanotechnology problems [14], heat transfer in two-phase media [11], and singularly perturbed diffusion problems with boundary and internal layers [13]. In an alternative approach, Eqs. (6) are replaced by an inverted diffusion functional [15, 16] for which they are the Euler–Lagrange equations. It was numerically shown in [17] that grids constructed by minimizing the diffusion functional are nearly orthogonal.

To construct coordinate transformations for generating grids with mesh refinement near the shoreline points $\mathbf{x}_i$ ($i=1,\ldots,N$) obtained by numerically solving Eqs. (3) at the nth time step, the monitor function $Z(\mathbf{x})$ was specified as

$$Z(\mathbf{x})=(\varphi(\mathbf{x}))^{\alpha}+\varepsilon,\quad \varphi(\mathbf{x})=\min_{i=1,\ldots,N}\rho(\mathbf{x},\mathbf{x}_i),\quad \alpha<0,\quad 0<\varepsilon\ll 1. \tag{7}$$

With the use of the inundation line found in the computation, a discrete coordinate transformation was constructed by solving Eqs. (6) for monitor function (7) with $\alpha=1.7$ and $\varepsilon=0.01$. Then Eqs. (3) were used to find a new inundation line, etc.

4. SOME NUMERICAL RESULTS

By applying the method described above, the inundation area in the Tohoku region (Japan, tsunami of the year 2011) was computed on curvilinear moving and fixed grids, as well as on a uniform rectangular grid in D by solving system (1). The wave height coming to the coast was set equal to 0.6 m. As was noted above, the task of primary interest in the tsunami runup problem is the determination of the inundation area, i.e., the limiting shoreline, which separates the water and the land maximally far away from the initial shoreline in the course of runup. Figure 3 shows the numerical results obtained on fixed and moving curvilinear 300×225 grids with in the (ξ_1,ξ_2) plane. (In the former case, we set $\partial x/\partial t=\partial y/\partial t=0$ in system (3).) The numerical results obtained on a uniform rectangular 1500×1200 grid in the plane (OX, OY) are depicted in Fig. 4. It can be seen that the limiting shoreline positions computed on the moving and fixed grids nearly coincide (the maximum difference in the shoreline positions is within 25 m).

Evidently, the inundation areas found on the curvilinear and rectangular grids differ little. The difference is caused primarily by the complex structure of the land topography, which was more accurately

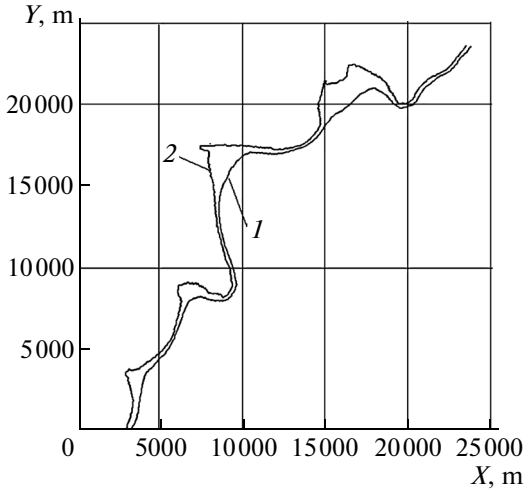

Fig. 3. Computation on the moving and fixed grids with 300 × 500 cells: (*1*) the initial shoreline and (*2*) final shoreline.

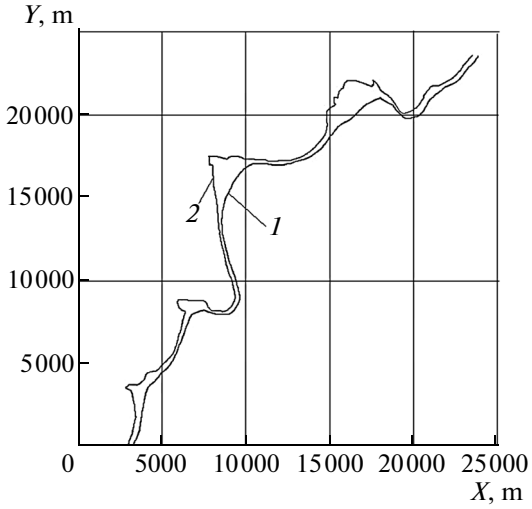

Fig. 4. Computation on a 1500 × 1200 rectangular grid in the physical domain *D*.

reproduced on the fine rectangular grid. It was found that the basic advantage of a moving grid is that the corresponding dynamics of the shoreline in the course of runup and rundown is computed much more accurately than on the fixed grid. However, the former case leads to a substantial increase in CPU time, because a moving grid has to be updated at every time step. Therefore, when the initial shoreline is not very jagged, the inundation area can be estimated using a fixed curvilinear grid, which has a rather small number of nodes, while ensuring acceptable numerical accuracy. Since the CPU time on a fixed curvilinear grid is roughly 1/40 times as much as that on a fine rectangular grid, curvilinear grids can be used in operational systems for estimating possible tsunami runup. The accuracy of the numerical solution was estimated by applying the conventional approach, i.e., computations on a sequence of refined meshes. As a result, the accuracy of the inundation area computed on the curvilinear (300 × 225) grid was estimated as roughly 5%.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research (project no. 13-01-00231) and by the Presidential Program "Leading Scientific Schools" (project no. NSh-5006.2014.9).

REFERENCES

1. V. L. Agoshkov, D. Ambrosi, V. Pennati, A. Quarteroni, and F. Saleri, "Mathematical and numerical modeling of shallow water flow," Comput. Mech. **11** (5–6), 280–299 (1993).
2. K. Kashiyama, Y. Ohba, T. Takagi, M. Behr, and T. Tezduyar, "Parallel finite element method utilizing the mode splitting and sigma coordinate for shallow water flows," Comput. Mech. **23** (2), 144–150 (1999).
3. A. Mohammadian, D. Y. Le Roux, and M. Tajrishi, "A conservative extension of the method of characteristics for 1D shallow flows," Appl. Math. Model. **31** (2), 332–348 (2007).
4. J.-S. Lai, W.-D. Guo, G.-F. Lin, and Y.-C. Tan, "A well-balanced upstream flux-splitting finite-volume scheme for shallow-water flow simulations with irregular bed topography," Int. J. Numer. Methods Fluids **62** (8), 927–944 (2010).
5. E. D. Fernandez-Nieto, J. Marin, and J. Monnier, "Coupling superposed 1D and 2D shallow-water models: Source terms in finite volume schemes," Comput. Fluids **39** (6), 1070–1082 (2010).
6. G. Oger, M. Doring, B. Alessandrini, and P. Ferrant, "Two-dimensional SPH simulations of wedge water entries," J. Comput. Phys. **213**, 803–822 (2006).
7. S. S. Khrapov, A. V. Khoperskov, N. M. Kuz'min, A. V. Pisarev, and I. A. Kobelev, "Numerical scheme for simulation of surface water dynamics based on the combined SPH–TVD approach," Vychisl. Metody Program. **12**, 282–297 (2011).

8. O. M. Belotserkovskii and Yu. M. Davydov, *Large Particle Method in Gas Dynamics* (Nauka, Moscow, 1982) [in Russian].
9. M. De Leffe, D. Le Touze, and B. Alessandrini, "SPH modeling of shallow-water coastal flows," J. Hydraulic Res. **48**, Extra Issue, 118–125 (2010).
10. A. V. Kofanov and V. D. Liseikin, "Grid construction for discretely defined configurations," Comput. Math. Math. Phys. **53** (6), 759–765 (2013).
11. A. V. Kofanov, V. D. Liseikin, and A. D. Rychkov, "Application of the spherical metric tensor to grid adaptation and the solution of applied problems," Comput. Math. Math. Phys. **52** (4), 548–564 (2012).
12. V. D. Liseikin, *Grid Generation Methods* (Springer, Berlin, 2010).
13. V. D. Liseikin, A. D. Rychkov, and A. V. Kofanov, "Applications of a comprehensive grid method to solution of three-dimensional boundary value problems," J. Comput. Phys. **230**, 7755–7774 (2011).
14. V. D. Liseikin, A. D. Rychkov, and A. V. Kofanov, *Adaptive Mesh Technology for the Numerical Solution of Applied Problems* (Novosibirsk. Gos. Univ., Novosibirsk, 2011) [in Russian].
15. N. T. Danaev, V. D. Liseikin, and N. N. Yanenko, "Numerical moving-grid computation of viscous gas flows past bodies of revolution," in *Numerical Methods in Continuum Mechanics* (Vychisl. Tsentr Sib. Otd. Akad. Nauk SSSR, Novosibirsk, 1980), Vol. 11, No. 1, pp. 51–61 [in Russian].
16. A. M. Winslow, *Adaptive Mesh Zoning by the Equipotential Method*, UCID-19062 (Lawrence Livermore National Laboratories, 1981).
17. V. A. Garanzha, "Barrier method for quasi-isometric grid generation," Comput. Math. Math. Phys. **40** (11), 1617–1637 (2000).

Translated by I. Ruzanova