Journal of the Oceanographical Society of Japan
Vol. 40, pp. 271 to 278, 1984

Numerical Simulation of Run-Up by Variable Transformation*

Hiroshi Takeda†

Abstract: A new method for the simulation of run-up using a variable transformation that fixes the shoreline is developed. This method uses equations expressed in the Eulerian description, but requires no artificial conditions at the shoreline. Hence, it may represent the real phenomenon more accurately than existing methods in which artificial conditions or extrapolation are needed. In a one-dimensional example the numerical solution is found to agree with analytic one very well. The method can easily be extended to two dimensions if the shoreline can be transformed into lines that intersect each other at right angles.

1. Introduction

In many numerical simulations of oceanic phenomena, some difficulty arises when the sea surface or a density interface intersects a slope. A typical example is tsunami run-up on dry land. Although many numerical simulations of tsunami have been attempted, they cannot easily be carried out because the shoreline is not fixed in space but moves with time.

There are three main methods to deal with such phenomena. The most commonly used is a method in which one approximates continuous topography by a series of discontinuous horizontal steps. This method requires an artificial condition at the shoreline which does not exist in the original equations. The methods by Aida (1977), Aida and Hatori (1982, 1983), Houston and Butler (1979), and Iwasaki and Mano (1979) belong to this category. The second is a method in which one extrapolates the velocity at the shoreline using the values at inner points. Hibberd and Peregrine (1979) used this method. These two methods, which are based on equations in the Eulerian description, avoid the computational difficulty at the shoreline by introducing an artificial assumption or approximation. Hence, we cannot have much confidence in their accuracy because computational errors most often occur at the shoreline. The third is a method based on equations expressed in the Lagrangian description, which was used by Goto and Shuto (1978, 1979). Although no artificial assumption is required at the shoreline in this method, it is necessary to derive long-wave equations and to consider viscosity in the equations, which are derived by the Eulerian approach.

In the present paper we introduce a method in which the shoreline is fixed by using a variable transformation. This method uses equations in the Eulerian description, but requires neither artificial assumptions nor extrapolation at the shoreline. Hence, we may anticipate that the present method may represent the run-up phenomena very accurately.

2. Basic equations in one-dimension

We use the long-wave equations governing hydrostatic motion of an inviscid, homogeneous, and incompressible water (with surface elevation η above its undisturbed level $z=0$) overlying the sea floor $z=-h$ (Fig. 1). Denoting the direction in which the sea extends by x, we

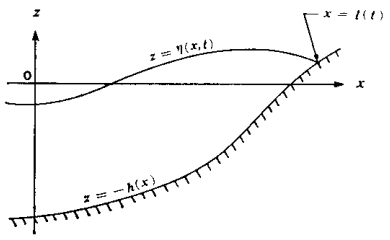

Fig. 1. Sketch of the one-dimensional case. The water overlies the sea floor $z=-h(x)$ and has a free surface at height η above its undisturbed level $z=0$.

* Received 17 November 1983; in revised form 19 March 1984; accepted 7 May 1984.

† Department of Applied Physics, Faculty of Engineering, University of Tokyo, Bunkyo-ku, Tokyo 113, Japan.

have the equations in one dimension:

$$\frac{Du}{Dt}=-g\frac{\partial\eta}{\partial x},$$
$$\frac{D\eta}{Dt}=-\frac{Dh}{Dt}-(h+\eta)\frac{\partial u}{\partial x}, \qquad (2.1)$$

where

$$\frac{D}{Dt}=\frac{\partial}{\partial t}+u\frac{\partial}{\partial x}$$

is the Lagrangian derivative (with time t and velocity in the x-direction u), and g is the acceleration due to gravity.

We transform the independent variables x and t by

$$X=\{l_0/l(t)\}x,$$
$$T=t,$$

where $l(t)$ is the shoreline position (Fig. 1) and $l_0=l(0)$. By this transformation, the region $0\leq x\leq l(t)$ is transformed onto the region $0\leq x\leq l_0$ which does not vary with time. The governing Eq. (2.1) are changed into

$$\frac{Du}{DT}=-gl'\frac{\partial\eta}{\partial X}, \qquad (2.2a)$$

$$\frac{D\eta}{DT}=-\frac{Dh}{DT}-l'(H+\eta)\frac{\partial\eta}{\partial X}, \qquad (2.2b)$$

where

$$\frac{D}{DT}=\frac{\partial}{\partial T}+U\frac{\partial}{\partial X}$$

with

$$U=l'u-(u_l/l)X,$$
$$l'=l_0/l$$

is the Lagrangian derivative in the X-T plane, and u_l is the value of u at the shoreline $x=l$ ($x=l_0$). In the X-T plane, the depth h and the shoreline position l are dependent variables, which are determined by the equations

$$H(X,T)=h(x)=h((l/l_0)X), \qquad (2.2c)$$
$$\frac{dl}{dT}=u_l, \qquad (2.2d)$$

where H denotes the depth in the transformed plane.

3. Finite difference scheme

We use the "path line method" (see Appendix A) in order to prevent nonlinear instability and apply the "forward-backward scheme" (see Appendix B) with respect to time.

Finite difference in space at the shoreline must be treated carefully. At the shoreline we cannot avoid the use of an uncentered difference for η when we predict u_l by using Eq. (2.2a). As a result, false reflection would occur at the boundary and the finite difference solution would soon diverge unless a special technique is introduced. False reflection occurs as a result of the fact that values previously computed by the uncentered difference are then used in the next step of calculation of space difference and a slight error in the former is magnified in the latter calculation leading to divergence (see Appendix C for details). Here, we adopt the meshes as shown in Fig. 2 to prevent false reflection. The meshes are staggered except at the shoreline $i=I$ (where i denotes the grid point), and at $i=I$ both u and η are specified. For these meshes we need not use $u_I(=u_l)$ in the space differences and hence, false reflection will not occur. (When the path line method is not used, the value of u_I is used in the calculation of the nonlinear term $(U\partial_X u)_{I-1}$. This, however, will hardly affect the solution because U is almost zero near the boundary.) We can determine η at the shoreline by letting it equal H (at $X=l_0$), which can be computed by Eqs. (2.2c) and (2.2d).

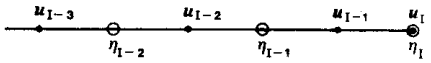

Fig. 2. Arrangement of grid points for u and η near the shoreline. Solid circles denote those for u and open circles those for η.

We further transform the coordinates in order to gain more accuracy near the shoreline:

$$X=R(\xi),$$
$$R(\xi)=(2-a)\xi+\{(a-1)/l_0\}\xi^2,$$

By this transformation, the mesh size ΔX decreases linearly as ξ increases (the mesh size $\Delta\xi$ is constant). Then we finally have the finite difference equations

$$l^{n+1}=l^n+\Delta T u_l{}^n\,,$$
$$H_i{}^{n+1}=h_0(R(\xi_i)/l'^{n+1})\,,$$
$$\eta_i{}^{n+1}=\eta_a{}^n-H_i{}^{n+1}+H_a{}^n-\Delta T l'^n(\eta_a{}^n+H_a{}^n)(\delta_\xi u)_a{}^n/(dR(\xi_a{}^n)/d\xi)\,,$$
$$u_i{}^{n+1}=u_a{}^n-\Delta T l'^{n+1}(\delta_\xi\eta)_i{}^{n+1}/(dR(\xi_i)/d\xi)\,,$$

where ΔT is the time interval and δ_ξ is the finite difference quotient corresponding to $\partial/\partial\xi$. For the space derivatives, the centered differences at the inner points are used, but at $i=I$, for example, the third-order uncentered difference is used:

$$(\delta_\xi\eta)_I=(11\eta_I-18\eta_{I-1}+9\eta_{I-2}-2\eta_{I-3})/(6\Delta\xi)\,.$$

4. A numerical example

Let us consider the case of a constant slope as an example and compare the numerical solution with the analytic one obtained by Carrier and Greenspan (1958). The distribution of surface elevation expressed by Eq. (3.1) in Carrier and Greenspan (1958) is adopted as the initial condition. The initial velocity is assumed to be zero everywhere. In the computaion, all the variables are non-dimensionalized after Carrier and Greenspan (1958). The time interval $\Delta t(=\Delta T)$ is 0.005, the grid points for η and u are both 51, and $a=0.1$. At the seaward boundary ($x=X=0$), the analytic solution is used.

Figure 3 shows the arrangement of grid points for η near the shoreline at $t=0$ to 0.5 at intervals of 0.1 with the elevation of the slope $h(x)=x$. Figure 4 is a comparison of the profiles of (a) η and (b) u near the shoreline at $t=0$ to 0.5 at intervals of 0.05 between the numerical (solid lines) and the analytic (dotted lines) solutions. Figure 5 shows the pofiles of (a) η and (b) u in the entire computational region at $t=$ 1.5 to 3.5 at intervals of 0.5 (dotted and solid line curves as in Fig. 4). Figure 6 shows time variation of the shoreline position and the velocity at the shoreline in both the computational (solid lines) and the analytic (dotted lines) solutions. These figures show that the numerical solution agrees quite well with the analytic one except in the vicinity of $t=4.5$ and 9 in Fig. 6, where the former oscillates slightly. This is because an error which was produced near the shoreline propagates towards the deep sea and is reflected at the seaward boundary

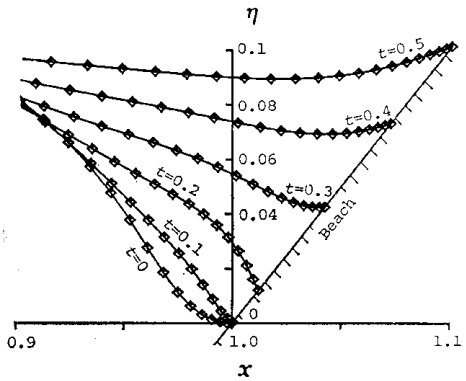

Fig. 3. Arrangement of grid points for surface elevation near the shoreline at $t=0$ to 0.5 at intervals of 0.1. Numbers of grid points for η and u are both 51, $\Delta t=0.005$ and $a=0.1$.

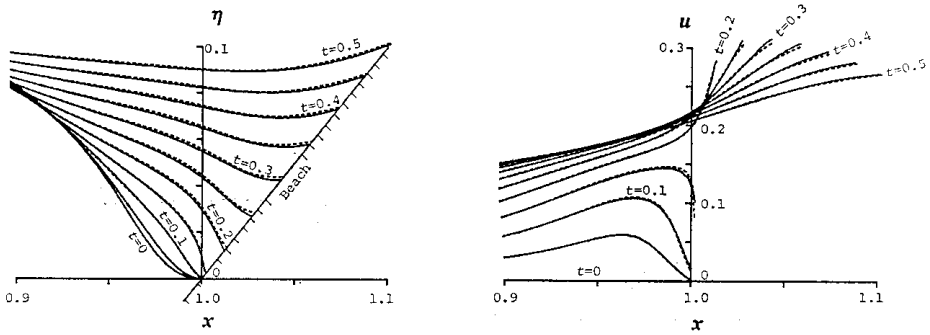

Fig. 4. Comparison of numerical solution (solid lines) with the analytic one (dotted lines) expressed by Eq. (3.1) in Carrier and Greenspan (1958) with $\varepsilon=0.2$. Distributions of surface elevation η (left) and velocity u near the shoreline (right) are plotted at $t=0$ to 0.5 at intervals of 0.05.

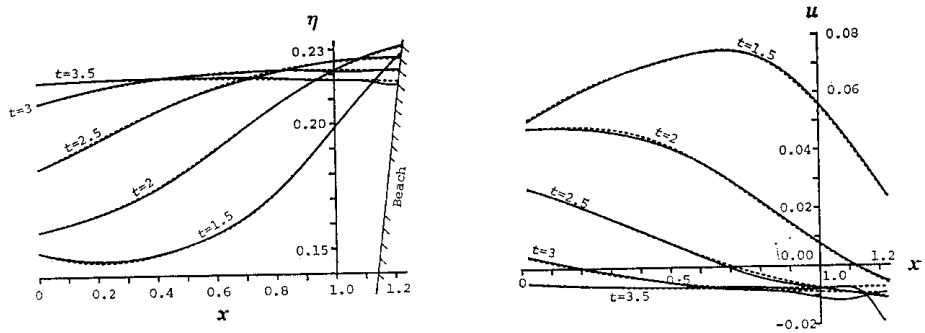

Fig. 5. Plot of the numerical (solid lines) and the analytic (dotted lines) solutions in the entire computational region at $t=1.5$ to 3.5 at intervals of 0.5.

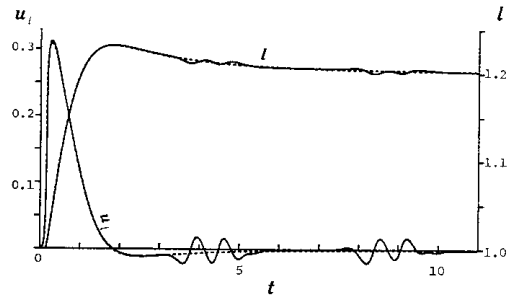

Fig. 6. Comparison of the shoreline variables (shoreline position l and shoreline velocity u_l) between the numerical and analytic solutions. The former is shown by solid lines and the latter by dotted lines.

where the analytic solution is used. We can actually see in Fig. 5 that an error with a small wavelength is reflected at the seaward boundary and propagates towards the shoreline. In the phenomena of run-up, the error, whose amplitude is small in the deep sea, may rapidly become larger as it propagates in the shallower sea. The magnitude of such error, of course, depends on the smoothness of the phenomena. Since the initial profile in this example was not smooth near the shoreline, a comparatively large error was produced. In simulations of actual phenomena, however, in which motions are usually smoother than this example and a viscous effect is present, the error due to reflection would be minimized and would not constitute a serious problem.

5. Application to two-dimensional problems

The method for one dimension can easily be applied to two dimensions if the shoreline can be transformed into lines that intersect each other at right angles. (The geometry shown in Fig. 7 corresponds to this case.) In this case, the transformation

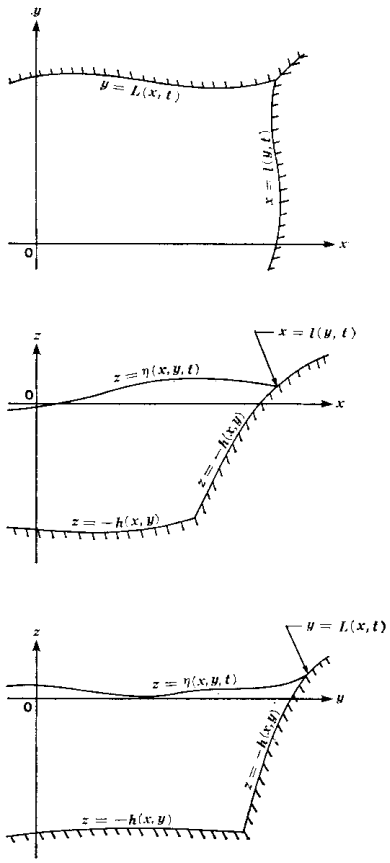

Fig. 7. Sketch showing the topography in two-dimensions.

$$X=l_0/l(y,t),$$
$$Y=L_0/L(x,t),$$
$$T=t,$$

results in the following derivatives:

$$\frac{\partial}{\partial x}=l'\frac{\partial}{\partial x}-\left\{l'Y\frac{\partial L}{\partial x}\Big/\frac{\partial(YL)}{\partial Y}\right\}\frac{\partial}{\partial Y},$$
$$\frac{\partial}{\partial y}=-\left\{L'X\frac{\partial l}{\partial Y}\Big/\frac{\partial(Xl)}{\partial X}\right\}\frac{\partial}{\partial X}+L'\frac{\partial}{\partial Y},$$
$$\frac{\partial}{\partial t}=\frac{\partial}{\partial T}-\frac{X}{lu_l}\frac{\partial}{\partial X}-\frac{Y}{Lv_l}\frac{\partial}{\partial Y},$$
$$\frac{D}{DT}=\frac{\partial}{\partial T}+U\frac{\partial}{\partial X}+V\frac{\partial}{\partial Y},$$

where

$$l'=l_0/l,$$
$$L'=L_0/L,$$
$$U=-\frac{X}{lu_l}+l'u-\left\{L'X\frac{\partial l}{\partial Y}\Big/\frac{\partial(Xl)}{\partial X}\right\}v,$$
$$V=-\frac{Y}{Lv_l}-\left\{l'Y\frac{\partial L}{\partial X}\Big/\frac{\partial(YL)}{\partial Y}\right\}u+L'v.$$

Using the above transformation, we can transform the governing equations and can solve them in a similar way to the one-dimensional case.

6. Conclusions and discussion

A new method for the simulation of run-up using a variable transformation that fixes the shoreline was developed. This method differs from existing ones in that it requires neither artificial assumptions nor extrapolations. The numerical solution by this method was found to agree well with the analytic one in a one-dimensional example. The present method can easily be extended to two dimensions if the shoreline can be transformed into lines that intersect each other at right angles.

General geometries can be handled if the transformation in Section 5 is used after mapping the region onto a rectangle by the "grid generation" technique (Thompson *et. al.*, 1974)[1]. If the form of the boundary varies rapidly in time, the computational region must be mapped onto the rectangle again during the calculation before the boundary form becomes double-valued.

[1] This idea was suggested by Dr. Motoyasu Miyata.

Although we have only considered a barotropic ocean so far, we can also apply the present method to a layered ocean in which the density interface intersects the shelf slope in a similar manner. In addition the method can also be applied to a continuosly stratified ocean and to the general equations in which the hydrostatic approximation is not valid, using the transformation that fixes the sea surface.

Acknowledgements

The author would like to express his great appreciation to Dr. M. Miyata and Prof. H. Takami for their careful reading of the manuscript and helpful comments.

References

Aida, I. (1977): Numerical experiments for inundation of tsunamis. Bull. Earthq. Res. Inst., **52**, 441-460 (in Japanese).

Aida, I. and T. Hatori (1982): Numerical experiments of tsunamis inundating Owase City, Central Japan. Bull. Earthq. Res. Inst., **57**, 337-350, (in Japanese).

Aida, I. and T. Hatori (1983): Numerical simulation of the tsunamis inundation in Ofunato City, Northeastern Japan. Bull. Earthq. Res. Inst., **58**, 175-185 (in Japanese).

Ames, W. F. (1969): Numerical methods for partial differential equations. Barnes and Noble, Inc., New York.

Carrier, G. F. and H. P. Greenspan (1958): Water waves of finite amplitude on a sloping beach. J. Fluid Mech., **4**, 97-109.

Fischer, G. (1959): Ein numerisches Verfahren zur Errechnung von Windstau und Gezeiten in Randmeeren. Tellus, **11**, 60-76.

Goto, C. and N. Shuto (1978): Numerical simulation of tsunami run-ups. Coastal Engg. in Japan, JSCE, **21**, 13-20.

Goto, C. and N. Shuto (1979): Two-dimensional numerical computation of nonlinear tsunami run-ups. Proc. of 26th Conf. on Coastal Engg., JSCE, 56-60 (in Japanese).

Hibberd, S. and D. H. Peregrine (1979): Surf and run-up on a beach: a uniform bore. J. Fluid Mech., **95**, 323-345.

Houston, J. R. and H. L. Butler (1979): A numerical model for tsunami inundation. U. S. Army Engineer Waterways Experiment Station, Tech. Report HL-79-2.

Iwasaki, T. and A. Mano (1979): Numerical computations of two dimensional run-up of tsunamis

due to the Eulerian description. Proc. 26th Japanese Conf. on Coastal Eng., 70-74 (in Japanese).

MacCormack, R. W. (1978): An efficient explicit-implicit-characteristic method for solving the compressible Navier-Stokes equations. SIAM-AMS Proc. Symp. Comput. Fluid Dyn., New York, 130-155.

MacCormack, R. W. and H. Lomax (1979): Numerical solution of compressible viscous flows. Ann. Rev. Fluid Mech., **11**, 289-316.

Mesinger, F. and A. Arakawa (1976): Numerical Methods Used in Atmospheric Models. Vol. 1, GARP Publications Series No. 17.

Thompson, J. F., F. C. Thames and C. W. Mastin (1974): Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. J. Comput. Phys., **15**, 299-319.

Appendix A. Path line method

The path line method is a modified version of the "characteristic method" (MacCormack, 1978; MacCormack and Lomax, 1979). In the former method, the time is integrated only along the path line in contrast to the latter approach in which it is integrated along all the characteristic paths. This simplified method is good enough to prevent nonlinear instability. The path line is determined from the equations

$$\int_{t^n}^{t^{n+1}} u dt = x_i - x_\alpha,$$

$$u_\alpha{}^n = \{u_k{}^n(x_{k+1}-x_\alpha)+u_{k+1}{}^n(x_\alpha-x_k)\}/\Delta x,$$

where the superscript n denotes the value at the n-th step and the subscript α denotes the value on the path line. The value of k is chosen so that the path line lies between x_k and x_{k+1} (Fig. 8). The integral $\int_{t^n}^{t^{n+1}} u dt$ is approximated in a suitable way. Here, we take $u_\alpha{}^n \Delta t$ (Δt is the time interval) at the first step and $(u_\alpha{}^n + u_i{}^{n+1,*})/(2\Delta t)$ at the second step,

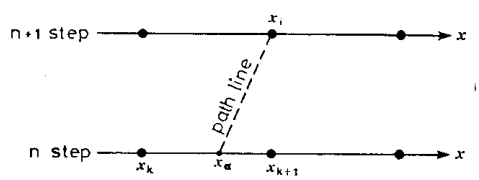

Fig. 8. Sketch showing location of path line between x_k and x_{k+1}.

where $u_i{}^{n+1,*}$ is a temporary value computed by using the finite difference equation for Eq. (2.2a) at the first step.

Appendix B. Forward-backward scheme

The forward-backward scheme (Fischer, 1959) is one in which the forward (Euler) scheme is applied to one of the two long-wave equations and after that, the backward scheme is applied to the other equation. In the present Eq. (2.2) where the depth h is also a dependent variable in the momentum equation, the forward scheme must be applied to the continuity equation.

The forward-backward scheme has the disadvantage that it is unstable when it is used for the advection term in nonlinear equations. However, such a problem does not arise if it is used in the "path line method" (Appendix **A**), because in that case the advection terms do not appear explicitly in the finite difference equations.

The forward-backward scheme is essentially the same as the leap-frog scheme. In addition, it is stable and neutral with time steps twice as large as those allowed by the CFL criterion for the leap-frog scheme (Ames, 1969). It becomes, therefore, a very good scheme if it is combined with the path line method. (See Mesinger and Arakawa (1976) for more detailed explanation about the forward-backward scheme.)

Appendix C. The cause of computational instability at the boundary

We will show that the error in the finite difference solution increases greatly through the space derivative term, and as a result computational instability may occur.

As an example we take the first-order wave equation:

$$\frac{\partial u(x,t)}{\partial t} = c\,\frac{\partial u(x,t)}{\partial x}, \qquad \text{(C1)}$$

where c is a constant. Denoting finite difference quotients for $\partial/\partial t$ and $\partial/\partial x$ by δ_t and δ_x and the difference solution by U, we have the finite difference equation for Eq.(C1):

$$(\delta_t U)_i{}^n = c(\delta_x U)_i{}^n. \qquad \text{(C2)}$$

Moreover, we define the error in the finite difference solution and the truncation errors in space and time by

$$E_i^n = U_i^n - u_i^n ,$$
$$\varepsilon^x{}_i^n = (\delta_x u)_i^n - (\partial u/\partial x)_i^n , \qquad \text{(C3)}$$
$$\varepsilon^t{}_i^n = (\delta_t u)_i^n - (\partial u/\partial t)_i^n ,$$

respectively. From Eqs. (C1) and (C3), we have

$$(\delta_t u)_i^n = c(\delta_x u)_i^n - c\varepsilon_i^n , \qquad \text{(C4)}$$

where

$$c\varepsilon_i^n = c\varepsilon^x{}_i^n - \varepsilon^t{}_i^n$$

is the truncation error in the difference equation.

We try to express the error in the solution E_i^n in terms of the truncation error ε. From Eqs. (C2), (C3) and (C4), we obtain

$$(\delta_t E)_i^n = c(\delta_x E)_i^n + c\varepsilon_i^n . \qquad \text{(C5)}$$

For simplicity, we use the Euler scheme for the time. Then we have, from Eq. (C5), the recurrence formula for E:

$$E_i^{n+1} = E_i^n + d(\delta_x E)_i^n + d\varepsilon_i^n , \qquad \text{(C6a)}$$

where

$$d = c\Delta t . \qquad \text{(C6b)}$$

Since E_i^0 are zero, we obtain the following equations for E_i^1, E_i^2, and E_i^3 from Eq. (C6a):

$$E_i^1 = d\varepsilon_i^0 ,$$
$$E_i^2 = d(\varepsilon_i^0 + \varepsilon_i^1) + d^2(\delta_x\varepsilon)_i^0 ,$$
$$E_i^3 = d(\varepsilon_i^0 + \varepsilon_i^1 + \varepsilon_i^2) + d^2(2(\delta_x\varepsilon)_i^0 + (\delta_x\varepsilon)_i^1) + d^3(\delta_x(\delta_x\varepsilon))_i^0 .$$

Hence, we assume the following expression for E_i^n:

$$E_i^n = \sum_{k=1}^{n} d^k \left\{ (\delta_x)^{k-1} \sum_{m=0}^{n-k} a_k{}^{n,m}\varepsilon^m \right\}_i , \qquad \text{(C7a)}$$

where

$$((\delta_x)^{k-1}\varepsilon^m)_i = (\delta_x \ldots (\delta_x(\delta_x\varepsilon^m))\ldots)_i ,$$
$$a_1{}^{n,m} = 1 , \qquad \text{(C7b)}$$
$$a_k{}^{n,n-k} = 1 .$$

On substituting Eq. (C7) into Eq. (C6a), we obtain, after some rearrangement,

$$E_i^{n+1} = \left[d\sum_{m=0}^{n}\varepsilon^m + \sum_{k=0}^{n} d^k(\delta_x)^{k-1} \left\{ \sum_{m=0}^{n-k} (a_{k-1}{}^{n,m} + a_k{}^{n,m})\varepsilon^m + \varepsilon^{n+1-k} \right\} + d^{n+1}(\delta_x)^n\varepsilon^0 \right]_i . \qquad \text{(C8)}$$

The condition that Eq. (C8) reduces to Eq. (C7) with $n \to n+1$ yields

$$a_k{}^{n+1,m} = a_{k-1}{}^{n,m} + a_k{}^{n,m}$$
$$(2 \le k \le n,\ 0 \le m \le n-k) . \qquad \text{(C9)}$$

Using Eq. (C7) with the relation (C9), we can estimate the magnitude of the error in the solution E in terms of the truncation error ε.

We further put $\varepsilon_i^m \simeq \varepsilon_i$ to simplify the discussion. Then Eqs. (C7) and (C9) reduce to

$$E_i^n = \left\{ \sum_{k=1}^{n} d^k A_k{}^n (\delta_x)^{k-1}\varepsilon \right\}_i , \qquad \text{(C10a)}$$

$$A_k{}^{n+1} = A_k{}^n + A_{k-1}{}^n$$
$$(n \ge 3,\ 2 \le k \le n-1) , \qquad \text{(C10b)}$$

where

$$A_k{}^n = \sum_{m=0}^{n-k} a_k{}^{n,m} . \qquad \text{(C10c)}$$

Since $A_1{}^n = n$ and $A_n{}^n = 1$ hold according to Eqs. (C7b) and (C10c), we obtain from the relation (C10b)

$$A_k{}^n = {}_nC_k , \qquad \text{(C11)}$$

where ${}_nC_k$ denotes the number of the combination.

If we use the centered difference at all grid points (on the assumption that the computational region is infinite), we can derive the equations

$$((\delta_x)^k\varepsilon^x)_i = 0(\Delta x^2) \quad (k = 0, 1, 2, \ldots) , \qquad \text{(C12)}$$

because the truncation error for the space derivative

$$(\varepsilon^x)_i = \frac{\Delta x^2}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_i + \frac{\Delta x^4}{5!}\left(\frac{\partial^5 u}{\partial x^5}\right)_i + \ldots$$

has the same coefficients at all the points. On the other hand, if we use the first-order uncentered scheme at the boundary point I, *i.e.*, if we assume

$$(\delta_x\varepsilon^x)_I = (\varepsilon^x{}_I - \varepsilon^x{}_{I-1})/\Delta x ,$$
$$\varepsilon^x{}_I = -\frac{\Delta x}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right)_I + \frac{\Delta x^2}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_I - \ldots$$

we can derive, after some calculations,

$$((\delta_x)^k\varepsilon^x)_I = O(\Delta x^{2-k})$$
$$(k=0, 1, 2, \ldots)\,. \qquad \text{(C13)}$$

The above equations show that the term $((\delta_x)^k\varepsilon^x)_I$ becomes very large for large k, compared with the case where the centered difference is used at all grid points. Since the product of d^k and $A_k{}^n$ expressed by Eqs. (C6b) and (C11) can also be large for large n and k, the error at the boundary $E_I{}^n$ in Eq. (C10a) will be enormous as n increases. As a result, the numerical solution will soon diverge. This is the essence of false reflection at the boundary. (It is noteworthy that computational instability at the outflow boundary results for the same reason.)

An alternative and more intuitive explanation of false reflection is as follows. A slight error generated near the boundary as a result of using a different kind of finite difference is enlarged when it is used in the following step of space difference calculation (*i.e.*, by dividing by the small value Δx).

Even when the second-order difference is used at the boundary, the above conclusion still holds. (This is the reason why accuracy is not improved and computational instability cannot be prevented even if the second-order uncentered difference is used at the outflow boundary.)

For the truncation error in the time derivative, we can derive the equations

$$((\delta_x)^k\varepsilon^t)_i = O(\Delta t) \quad (k=0, 1, 2, \ldots)\,,$$
$$((\delta_x)^k\varepsilon^t)_I = O(\Delta t \Delta x^{1-k}) \quad (k=1, 2, 3, \ldots)$$

corresponding to Eqs. (C12) and (C13), respectively. These equations show that the error $E_I{}^n$ also becomes greater because of the truncation error in the time derivative.

The above result leads to the conclusion that computational instability at the boundary can be prevented if the value calculated by the uncentered difference is not used in the calculation of any space derivatives in the following step.

変数変換による Run-Up シミュレーション

竹　田　　宏*

要旨: 海岸線を固定するような変数変換を用いることにより run-up のシミュレーションを行なおうとする新しい手法が提示された．この方法は，オイラー的に記述された方程式系を用いるにもかかわらず，海岸線においていかなる人為的な仮定をも必要としない．したがって，人為的条件，あるいは外挿を必要とする従来の方法に比べて，より実際の現象を正確に再現し得ることが期待される．実際，一次元における計算例において，数値解と解析解は非常によい一致を示す．また，この方法は，海岸線が互いに直交する直線に変換される場合には，容易に二次元に拡張することができる．

* 東京大学工学部 物理工学科
〒113 東京都文京区本郷 7-3-1