INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 4, 1137–1148 (1984)

A KANTOROVICH COMPUTATIONAL METHOD FOR FREE SURFACE GRAVITY FLOWS

E. F. TORO

Department of Applied Mathematical Studies, University of Leeds, Leeds LS2 9JT, England

AND

M. J. O'CARROLL

Department of Mathematics, Teesside Polytechnic, Middlesbrough, Cleveland TS1 3BA, England

SUMMARY

Computing free surface gravity flows involves basically two coupled problems, namely, the location of the free surface position and the determination of the internal flow field (for assumed values H_0 and Q of the total head and discharge, respectively).

Solution techniques are invariably based on iterative procedures, but those that iterate between the two coupled problems may become unstable.

In this paper we present a computational method in which the coupling is kept throughout the process of iteration. This is achieved by converting the coupled problems (by means of the Kantorovich method) into the single problem of finding a set of streamlines, including that of the free surface. These streamlines are moved (iteratively) to satisfy the stationary conditions of the governing variational principle.

The algorithm is very stable and converges rapidly. It is also easy to implement to solve various types of steady flows with a free surface under gravity.

KEY WORDS Free Surface Flows Computation Kantorovich Method

1. INTRODUCTION

Computational techniques for free-surface gravity flows continues to be an active area of research. The problem is not a simple one even when neglecting surface tension and viscosity and assuming flows to be steady and irrotational. In this paper we are concerned with this restricted version of the problem. Yeung[1] has given an up to date account of some of the most popular methods employed in computing free surface flows in the sense adopted here.

The distinguishing feature of free-surface flows is that the position of the free surface is not known *a priori*. Therefore, the flow domain itself is unknown. This problem constitutes the difficult part and is non-linear. The other problem is the determination of the internal flow field. This is an easy problem (linear) and consists of solving Laplace's equation subject to appropriate boundary conditions. The solution to these coupled problems is usually accomplished by handling the non-linear free-surface condition in a variety of iterative techniques.

The finite difference method was applied by Southwell and Vaisey[2] to compute various examples of free-surface flows. Theirs was the first reported attempt to solve these problems numerically. A

0271–2091/84/12 1137–12$01.20

Received 2 August 1983
Revised 3 January 1984

popular technique in more recent times is the finite element method. This has been used to compute flows over a spillway,[3,4] critical flows over weirs,[5-7] and water waves[6,8] etc. Boundary integral techniques have also been applied to compute surface problems. Their merits stem from the fact that iterative procedures are needed only on the free surface, without explicit coupling to the internal flow problem. Byatt-Smith and Longuet-Higgins[9] used this technique to compute solitary waves.

Improved techniques based on power series expansions have been used by Cokelet to compute water waves.[10] His numerical results are reported to be very accurate for a certain range of waves. Cokelet's technique makes use of Pade approximants, which extend beyond the radius of convergence of ordinary power series. The use of Pade approximants in free-surface problems was introduced by Schwartz[11] in 1974 who was concerned with the study of water waves.

Hodograph transformations have been used in the past to solve some free-surface problems of the type we are concerned with here, but the success of this technique was confined to problems with small gravitational effects. Quite recently however, Han and Chow[12] have successfully applied this method to compute more general free surface gravity flows.

In this paper we introduce the application of the Kantorovich method[13] to solve free-surface problems numerically. The theoretical aspects of the Kantorovich method are found in Reference 13. The principal objective of this paper is to present a detailed description of the derived numerical technique and to illustrate its implementation for practical applications. A validation of the method has been carried out by the authors and details of comparison with other numerical, theoretical and experimental data are reported in References 6 and 7.

The present model starts from a formulation of the problem in terms of a stream function ψ and a variational principle with functional J. The coupled problems of finding the free surface and the internal flow field are converted into the single problem of finding a set of N streamlines $y_i(x) (i = 1, \ldots, N)$ to make J stationary. The solid boundary defining the bed profile and the free surface are both streamlines and we denote them by $y_0(x)$ and $y_N(x)$, respectively. The stream function ψ takes on constant values Q_i on each streamline $y_i(x)$. Since ψ is to take on the value zero on the free surface and the value of the discharge Q on the bed (or vice versa), a natural choice for Q_i is $Q(N-i)/N$. That is to say, the interval $[0, Q]$ has been discretized by $N+1$ equally spaced points Q_i into N subintervals. ψ is specified on each streamline. Within the stream layers we introduced an approximation whereby ψ is interpolated to be linear in y. This is the first semidiscrete approximation of the problem and it remains continuous in x. Stationary conditions on J yield a system of non-linear ordinary differential equations for the unknown streamlines $y_i(x)$ $(i = 1, \ldots, N)$.

In Section 4 we approximate the problem further by expressing derivatives in terms of finite differences. In this way the system of non-linear ordinary differential equations is reduced to a system of $N \times M$ non-linear algebraic equations. Considerable attention has been paid here to the efficiency with which these algebraic equations are solved. Obviously, the use of any library routine may prove satisfactory for practical applications. But, as reported in Reference 6, large problems become very demanding in computing time.

Here we achieved efficiency by (i) deriving the stationary equations analytically, (ii) differentiating the stationary equations exactly to determine their Jacobian matrix, (iii) using the banded structure of the Jacobian matrix and (iv) carefully programming the algebra. A Newton–Raphson iteration procedure is used to solve the non-linear system. The algorithm is described in Section 5 and it is found to be very stable and efficient in computing time even for near-critical flows.[6] Its double-precision implementation would normally require about six iterations to terminate with both the residuals in the stationary equations and the maximum movement of the streamlines becoming less than 10^{-14}.

In Section 6, as an illustration, we have included some wave computations. These are not aimed at analysing water-wave characteristics, but simply as an example of application of the method.

2. STATEMENT AND FORMULATION OF THE PROBLEM

We consider non-viscous flows with a free surface under gravity which are steady, two-dimensional, incompressible and irrotational. In terms of a volumetric stream function $\psi(x, y)$ both the bed (a fixed boundary) and the free surface are streamlines. A typical flow domain is shown in Figure 1, where H_0 measures the stagnation level (total head), L gives the length of the flow region in the x-direction, $b(x)$ is the position of the prescribed bed profile and $h(x)$ measures the depth of flow, which is to be determined.

Under the stated physical assumptions and appropriate boundary conditions we obtain the following boundary value problem:

$$\left.\begin{aligned}&\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=0 \quad \text{in} \quad R\\ &\psi=Q \quad \text{on} \quad \text{BD}\\ &\psi=0 \quad \text{on} \quad \text{FS}\\ &\frac{\partial\psi}{\partial n}=0 \quad \text{on} \quad \text{BF} \quad \text{and} \quad \text{DS}\end{aligned}\right\} \tag{1}$$

$$\tfrac{1}{2}\left(\frac{\partial\psi}{\partial n}\right)^2+gy=0 \quad \text{on} \quad \text{FS} \tag{2}$$

where Q denotes the discharge and g the acceleration due to gravity.

The boundary conditions at the inlet and outlet boundaries correspond to normal flow and are adequate for most applications. The second (non-linear) boundary condition (2) on the free surface FS is derived from Bernoulli's law under the assumption of atmospheric pressure.

Equation (1) gives the flow field for an assumed position of FS, and prescribed values for the total head H_0 and the discharge Q. We call this the ψ-problem. Equation (2) provides a condition for finding the correct position of the unknown boundary FS, determined by $h(x)$. We term this the H-problem.

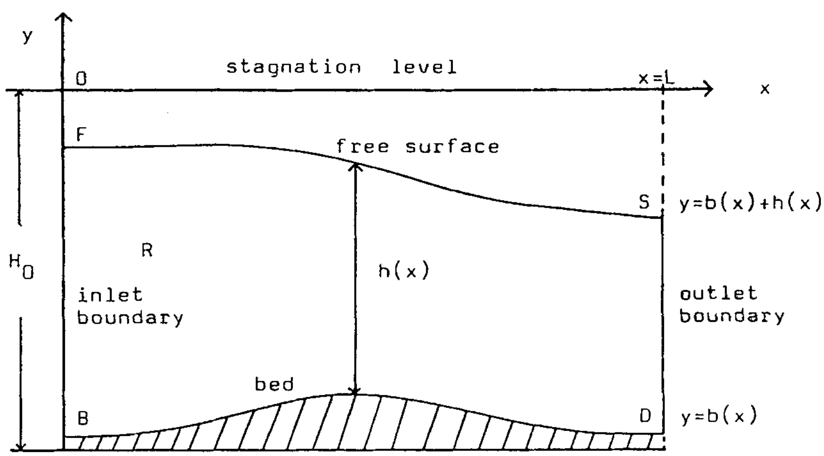

Figure 1. Flow domain with a free surface under gravity

In some situations the parameters of the problem H_0 and Q become themselves unknowns to be determined e.g. critical flows over weirs.[5–7]

The free boundary value problem (1), (2), which governs the combined H–ψ problems, is equivalent to the non-dimensionalized variational problem[14] with functional

$$J(h(x), \psi(x, y)) = \int_0^L \int_{b(x)}^{b(x)+h(x)} [\tfrac{1}{2}(\nabla\psi)^2 - y]\, \mathrm{d}x\mathrm{d}y \tag{3}$$

together with the constraints

$$\psi = Q \text{ on BD and } \psi = 0 \text{ on FS} \tag{4}$$

This means that for given H_0 and Q a solution of the combined H–ψ problems is a stationary point of the functional J. In expressions (3) and (4) all quantities have been non-dimensionalized with respect to length H_0 and time $(H_0/g)^{1/2}$, including the x and y co-ordinates.

The variational formulation as given by (3) and (4) is the starting point for direct finite element discretizations as in References 3–6 and 11 etc.

3. DERIVATION OF THE SEMIDISCRETE EQUATION BY THE KANTOROVICH METHOD

The Kantorovich method, or semidiscrete method, is a generalization of the Ritz method and is similar to the method of lines and its generalization, the method of discretization in time.[15]

In this section we apply the Kantorovich method directly to the problem as formulated in Section 2 in terms of a stream function and a variational principle (equations (3) and (4)).

From equation (4) we have $\psi = Q$ (the discharge) on the bed and $\psi = 0$ on the free surface. We begin by subdividing the interval $[0, Q]$ into a number N of subintervals $I_i = [Q_{i-1}, Q_i]$ of equal length Q/N, corresponding to stream layers. That is

$$Q_{i-1} = Q(N - i + 1)/N \quad \text{and} \quad Q_i = Q(N - i)/N, \text{ for } i = 1, \ldots, N$$

To each number Q_i we associate a streamline $y_i(x)$ on which the stream function $\psi(x, y)$ takes on the constant value Q_i (see Figure 2). In this way we determine N stream layers R_i associated with I_i and defined as follows:

$$R_i = \{(x, y), \text{ such that } 0 \leqslant x \leqslant L;\ y_{i-1}(x) \leqslant y \leqslant y_i(x)\}$$

No approximation has been imposed so far. The bed and free surface profiles are still the streamlines of the original flow domain.

Now we introduce our first approximation by defining the stream function $\psi(x, y)$ on each subdomain R_i as follows

$$\psi_i(x, y) = Q\{y_{i-1}(x) - y + h_i(x)(N - i + 1)\}/Nh_i(x), \text{ for } i = 1, \ldots, N \tag{5}$$

so that

$$\psi(x, y) = \begin{cases} \psi_i(x, y), \text{ for } (x, y) \text{ in } R_i \\ 0, \text{ otherwise} \end{cases}$$

Notice that the required boundary conditions are satisfied automatically and that for each streamline we have

$$y_i(x) = \sum_{j=0}^{i} h_j(x)$$

with $h_0(x)$ representing the bed elevation (fixed) above a horizontal datum $y = -1$.

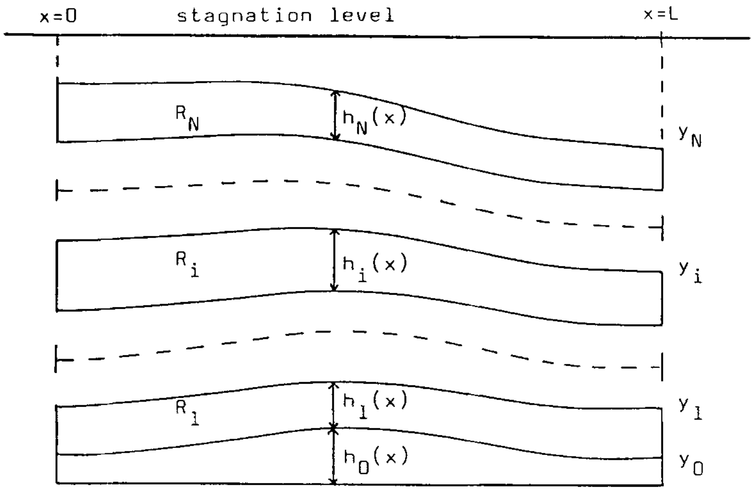

Figure 2. Flow domain subdivided into N stream layers R_i

As illustrated in Figure 2, each stream layer R_i has unknown thickness $h_i(x)$. The unknown streamlines $y_1(x),\ldots,y_N(x)$ are determined by the functions $h_1(x),\ldots,h_N(x)$. At this stage we emphasize that the problem of finding $\psi(x,y)$ and $h(x)$ as defined in Section 2, has been transformed into the problem of finding the set of functions $h_1(x),\ldots,h_N(x)$. These functions are to satisfy a system of non-linear (ordinary) differential equations, as we shall see later.

Obviously, the points Q_i in the interval $[0,Q]$ need not be taken equally spaced. Similarly, the approximation for the stream function ψ defined by equation (5) need not be linear in y on each subdomain R_i, but may be of higher order.

Next, we evaluate the functional J in terms of the admissible functions ψ_i. From equation (5) we obtain

$$\left.\begin{aligned}\partial\psi_i/\partial x &= Q\{h_i y'_{i-1} - h'_i(y_{i-1}-y)\}/Nh_i^2\\ \\ \partial\psi_i/\partial y &= -Q/Nh_i\end{aligned}\right\}\tag{6}$$

where for convenience the arguments x and y have been dropped and a prime denotes differentiation with respect to x.

The functional J, evaluated over the stream layer R_i, may be expressed as

$$J(R_i) = \int_0^L F_i(x,h_i,h'_i)\,\mathrm{d}x$$

where

$$F_i(x,h_i,h'_i) = \int_{y_{i-1}}^{y_i} \{\tfrac{1}{2}(\nabla\psi_i)^2 - y\}\,\mathrm{d}y$$

By substituting the partial derivatives given by equation (6) into $F_i(x,h_i,h'_i)$ we obtain

$$F_i(x,h_i,h'_i) = Q^2(3+3y'^2_{i-1}+3y'_{i-1}h'_i+h'^2_i)/(6N^2h_i^2) - (2y_{i-1}+h_i)h_i/2 \tag{7}$$

By setting

$$h=(h_1,\ldots,h_N);\ h'=(h'_1,\ldots,h'_N)$$

and

$$F(x,h,h')=\sum_{j=1}^{N}F_j(x,h_j,h_j')$$

the functional J evaluated over the entire flow domain becomes

$$J(x,h,h')=\int_0^L F(x,h,h')\,\mathrm{d}x \tag{8}$$

The solution to the problem as stated in Section 2 is a stationary point of the functional J, which satisfies the Euler–Lagrange equations

$$G_i\equiv\partial F/\partial h_i-\frac{\mathrm{d}}{\mathrm{d}x}(\partial F/\partial h_i')=0,\ \text{for } i=1,\dots,N$$

Some algebraic manipulations lead to the following system of ordinary differential equations for the unknown functions $h_1(x),\dots,h_N(x)$

$$G_i=\frac{-Q^2}{6N^2}\Bigg\{(2h_ih_i''+3h_iy_{i-1}''+3y_{i-1}'^2+3-h_i'^2)/h_i^2$$
$$+3\sum_{r=i+1}^{N}\{(h_rh_r''-h_r'^2+2(h_ry_{r-1}''-h_r'y_{r-1}'))/h_r^2\}\Bigg\}-y_N \tag{9}$$

together with the boundary conditions

$$h_i'(0)=h_i'(L)=0,\ \text{for } i=1,\dots,N \tag{10}$$

The problem as expressed by equations (9) and (10) remains continuous in x. At this intermediate stage between the original problem and the final numerical results some of the features of the Kantorovich method suggest themselves. One may explore the possibility of finding exact solutions to (9) and (10) or analysing some characteristics of the stationary point before arriving at numbers. However, such work is beyond the scope of the present paper. Nevertheless, it is interesting to note that the special case $N=1$ constitutes a shallow water type of approximation[6] for which solutions can be given explicitly in terms of cnoidal functions.

4. DERIVATION OF THE DISCRETE NON-LINEAR EQUATIONS AND THEIR JACOBIAN

The algorithm presented in this paper is based on the numerical solution of the non-linear boundary value problem (9), (10). We call the algorithm NODE, which stands for Non-linear Ordinary Differential Equations.

In order to solve (9) and (10) we first approximate derivatives in (9) by finite differences in the usual manner. In this way the differential problem (9), (10) is reduced to a system of algebraic (non-linear) equations. It is these discrete equations which we eventually solve numerically.

In this section we indicate the derivation of the algebraic equations and their corresponding Jacobian matrix, which are given explicitly in the Appendix.

The domain for equations (9) is discretized by M equally spaced points $x_j(j=1,\dots,M)$. We set $c=h_{j+1}-h_j$, $h_{ij}=h_i(x_j)$ and use second order central differences for first and second derivatives and fictitious nodes x_0 and x_{M+1} with boundary conditions

$$h_{i0}=h_{i2};\ h_{iM-1}=h_{iM+1},\quad \text{for}\quad i=0,\dots,N \tag{11}$$

After some algebra we obtain an expression for the general stationary equation (discrete) $G_{ij} = 0$ for $j = 1, \ldots, M$ and $i = 1, \ldots, N$. (Equations (12) in the Appendix).

Equations (12) are a system of non-linear algebraic equations to be solved for $N \times M$ unknowns h_{ij}. Given a flow-domain length L and a bed profile determined by $h_{0j}(j = 1, \ldots, M)$ we need to prescribe the mesh size $N \times M$ and the discharge Q.

In Reference 6 we used a general library subroutine due to Brown[16] and this proved satisfactory for most applications considered there, although it was not designed for the present purpose. For large problems the computing time became prohibitively long. The present version of the algorithm is efficient in terms of storage requirements and computing time.

We solve equations (12) by a Newton–Raphson iteration procedure where the entries of the Jacobian matrix S are derived analytically. This gives significant savings in computing time compared with numerical evaluation of derivatives from G_{ij}-function evaluations. In addition advantage is taken of the banded structure of the Jacobian matrix in the direct solution for each Newton–Raphson iteration.

As evident in (12) and in Figure 3, a function G_{ij} involves at most $3N$ variables, namely h_{kj-1}, h_{kj} and h_{kj+1} (for $k = 1, \ldots, N$). At the inlet and outlet boundaries only $2N$ variables are involved after using boundary conditions (11). Thus the Jacobian matrix S has block tridiagonal form with blocks of order $N \times N$ and bandwidth $4N - 1$. The coupling between columns of variables h_{ij} in Figure 3 gives full block matrices on the sub- and super-diagonals. We use a fixed bandwidth solver for bandwidth $4N - 1$ without exploiting zeros within that bandwidth since there is little gain in doing so.

It would be possible to obtain a system of equations with only 9-point coupling by working in node positions y_{ij} instead of stream layer thicknesses h_{ij}. This would save in assembly time and storage, but the resulting bandwidth would still be $2N + 3$ so that solution time saving would be limited. It is not clear how this would affect the performance and stability of the method for the non-linearities.

In the present method there are nine types of equations for the elements of the Jacobian matrix S. Three cases come from considering all variables lying below layer i at stations $j - 1, j$ and $j + 1$.

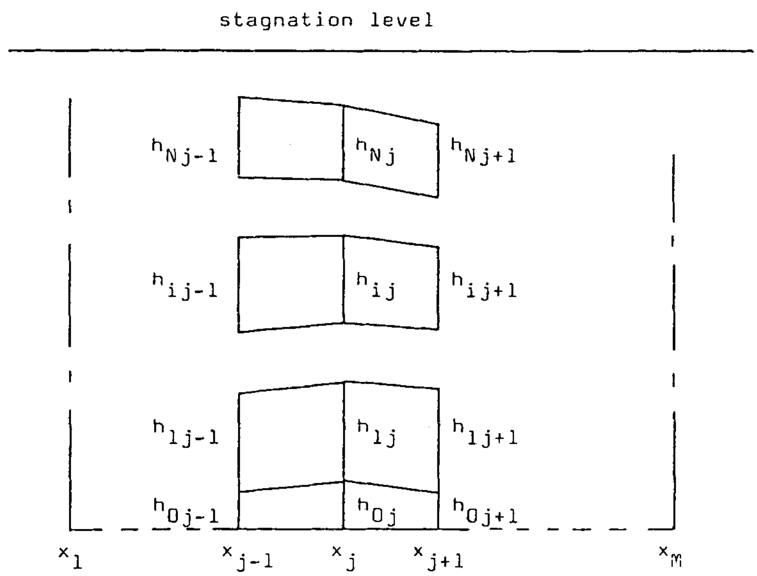

Figure 3. Discretized flow domain expressed in terms of variables h_{ij} ($j = 1, \ldots, M$; $i = 1, \ldots, N$)

Another three cases arise from layer i and finally three more cases come from variables above layer i.

The nine types of equations for the entries of the Jacobian matrix are given explicitly in the Appendix (equations (13)).

5. SOLUTION OF THE NON-LINEAR ALGEBRAIC EQUATIONS

The system of equations (12) is solved here by a Newton–Raphson type of iteration, whereby at iteration $k+1$ we solve a linear system of equations

$$S^k\delta = -G^k \tag{14}$$

for a vector δ of corrections δ_{ij} to h_{ij}.

S^k and G^k denote the Jacobian matrix and the residuals in the stationary equations evaluated at the previous iteration number k. The corrected value for the vector H of unknowns h_{ij} is given as

$$H^{k+1} = H^k + \delta \tag{15}$$

The iteration procedure as given by equations (14) and (15) proceeds as illustrated by the

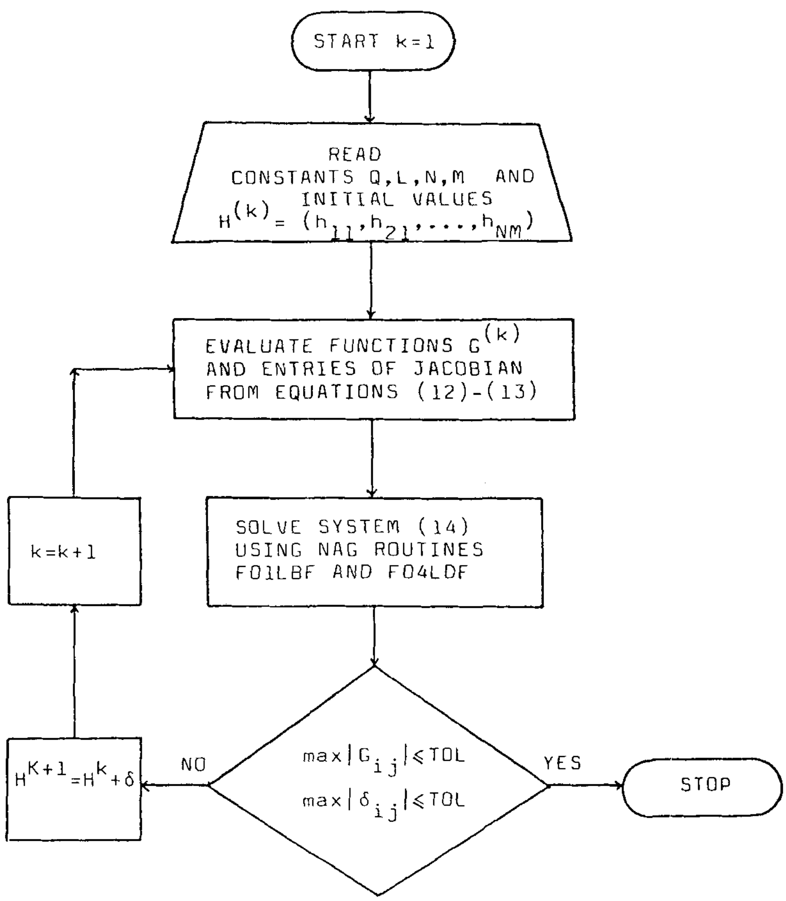

Figure 4. Algorithm for solving the algebraic equations (12)

diagram of Figure 4. It takes normally about six iterations to obtain a converged solution. The process is stopped when both the function residuals and the maximum streamline movement are less than a preassigned value for the tolerance TOL. In all computations we took TOL $= 10^{-14}$

The resulting computational method performs very well when applied to real problems. In Reference 6 reference was made to the remarkable stability of the method when computing critical flows and large-amplitude water waves. This feature is confirmed here. The algorithm (in double precision) is programmed in FORTRAN 77 in the AMDAHL V/7 computer of the University of Leeds. As an indication of the computing time required for this algorithm, a large amplitude water wave in the short wave region using a 10×8 mesh took 1·2 s starting from a sinusoidal initial surface profile. Previous calculations for the same problem on a UNIVAC 1110 using only a general purpose non-linear solver took some 3000 times longer.

6. AN ILLUSTRATIVE EXAMPLE: WAVES

For the purpose of illustrating a relatively simple application of NODE, we consider here the problem of computing waves over a horizontal bed. Surface gravity waves of permanent form on water of finite depth, when considered in a frame of reference that is fixed relative to a wave crest, may be represented in the present model of steady ideal flow.

Here we choose an example which has already been computed by Southwell and Vaisey.[2] Its computational aspects are very straightforward indeed, which is particularly helpful in highlighting the features of the present method.

The aim is to compute waves on a stream of fixed depth $h = 11/12$. That is to say, a constant depth h and velocity V are postulated for the undisturbed stream. The uniform flow relation

$$Q^2 = 2h^2(1 - h)$$

gives the corresponding value $Q^2 = 0{\cdot}1400463$.

Other data values to be provided are the channel length L and the mesh size $N \times M$ and the initial values for the streamlines. Symmetry allows only half a wave to be considered. From the boundary conditions being imposed, we expect to compute waves with trough at the inlet and crest at the outlet, or vice versa.

The prescribed channel length L is related to the flow rate Q (or equally to the undisturbed depth h) and to the amplitude of the wave to be computed. Any arbitrary choice of L will generally produce only the uniform flow solution of depth h. Small amplitude waves have wavelength λ_0 related to h by the linearized theory:

$$\lambda_0 = 4\pi(1 - h)/\tanh(2\pi h/\lambda_0) \tag{16}$$

As the amplitude increases, so the wavelength decreases (in the short wave region as for this example). By prescribing values of L slightly below $\lambda_0/2$ it is possible to compute finite-amplitude waves. Values of L slightly above $\lambda_0/2$ may give computed waves on a coarse mesh, but refinement restricts solutions to the case $L \leqslant \lambda_0/2$. Solving (16) iteratively gives $\lambda_0 = 1{\cdot}047$ in the case considered. In Figure 5 we illustrate a computed wave for $\lambda = 0{\cdot}93$ ($L = 0{\cdot}465$). The computed amplitude is $A = 0{\cdot}0711$. Here we take A as the displacement of the free surface at the crest above the position of the undisturbed stream h. The mesh used in all computations reported here was 15×10 and total CPU time was about 4 s for each wave starting from a sinusoidal surface.

From a large number of computations we have observed that the algorithm is capable of computing waves of practically all realistic amplitudes. Since the amplitude emerges as a result of computation, a check against the linear theory (see equation (16)) can be carried out by curve fitting

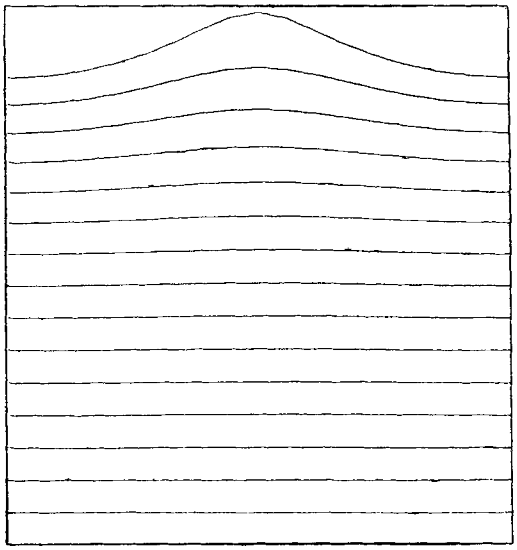

Figure 5. Computed wave for $\lambda = 0{\cdot}93$ and $h = 11/12$

in the (λ, A)-plane. This technique gives a limiting solution $\lambda_0 = 1{\cdot}047$ correctly to three decimal places. The computed solution of Southwell and Vaisey[2] is $\lambda_0 = 1{\cdot}11$, which is about 6 per cent higher than our computed value and the theoretical solution.

7. CONCLUSIONS

A method has been presented for the computation of steady ideal flows with a free surface under gravity. The method determines the flow by moving streamlines to satisfy a variational formulation of the flow conditions including the free-surface pressure condition.

The method has been thoroughly validated against other numerical and experimental results for the case of critical flows over weirs and against linearized wave theory, particularly in Reference 6. In this paper the method is presented in an improved form in respect of computational efficiency.

Advantages which the method offers include robust stable performance in calculating large-amplitude waves and critical flows for which other methods have encountered difficulties. The method is also easily implemented and efficient, requiring only a few seconds of computing time for problems of about 170 nodes, or about a minute for about 900 nodes, for strongly non-linear problems.

The program is applicable to a range of problems with free surfaces under gravity and arbitrary bed configuration. The method could also be adapted for other free-surface flow problems such as those with cavitation or jets.

ACKNOWLEDGEMENTS

The first named author is indebted to the Mathematics Department, Teesside Polytechnic and the Department of Applied Mathematical Studies, University of Leeds for the financial and academic support provided that contributed to the completion of this work.

APPENDIX

Here we give the stationary equations G_{ij} and the entries of the Jacobian matrix $\partial G_{ij}/\partial h_{km}$

$$G_{ij} = C(A_{ij} + B_{ij}) - y_{Nj} = 0, \quad i = 1,\ldots,N; j = 1,\ldots,M \tag{12}$$

with $C = -Q^2/(24c^2N^2)$ and

$$\begin{aligned}
A_{ij} &= \frac{8h_{ij}(h_{ij+1} - 2h_{ij} + h_{ij-1}) - (h_{ij+1} - h_{ij-1})^2 + 12c^2}{h_{ij}^2} \\
&\quad + 3\left\{\frac{(y_{i-1j+1} - y_{i-1j-1})^2 + 4h_{ij}(y_{i-1j+1} - 2y_{i-1j} + y_{i-1j-1})}{h_{ij}^2}\right\} \\
B_{ij} &= 3\sum_{r=i+1}^{N}\left\{\frac{4h_{rj}(h_{rj+1} - 2h_{rj} + h_{rj-1}) - (h_{rj+1} - h_{rj-1})^2}{h_{rj}^2}\right. \\
&\quad + 8\left(\frac{y_{r-1j+1} - 2y_{r-1j} + y_{r-1j-1}}{h_{rj}}\right) \\
&\quad \left. - 2\left\{\frac{(h_{rj+1} - h_{rj-1})(y_{r-1j+1} - y_{r-1j-1})}{h_{rj}^2}\right\}\right\}
\end{aligned}$$

The Jacobian entries are given as $S_{km}^{ij} = \partial G_{ij}/\partial h_{km}$, where the range of values for k is $1 \leqslant k < i$ for the first 3 equations. $k = i$ for the following 3 and $i < k \leqslant N$ for the last 3 equations.

$$\left.\begin{aligned}
S_{kj-1}^{ij} &= 6C\left\{\frac{2h_{ij} - y_{i-1j+1} + y_{i-1j-1}}{h_{ij}^2} + \sum_{r=i+1}^{N}\left(\frac{4h_{rj} + h_{rj+1} - h_{rj-1}}{h_{rj}^2}\right)\right\} \\
S_{kj}^{ij} &= -24C\left\{1/h_{ij} + 2\sum_{r=i+1}^{N}(1/h_{rj})\right\} - 1 \\
S_{kj+1}^{ij} &= 6C\left\{\frac{2h_{ij} + y_{i-1j+1} - y_{i-1j-1}}{h_{ij}^2} + \sum_{r=i+1}^{N}\left(\frac{4h_{rj} - h_{rj+1} + h_{rj-1}}{h_{rj}^2}\right)\right\} \\
S_{ij-1}^{ij} &= 2C\left\{\frac{4h_{ij} + h_{ij+1} - h_{ij-1}}{h_{ij}^2} + 3\sum_{r=i+1}^{N}\left(\frac{4h_{rj} + h_{rj+1} - h_{rj-1}}{h_{rj}^2}\right)\right\} \\
S_{ij}^{ij} &= 2C\left\{\frac{(h_{ij+1} - h_{ij-1})^2 - 3(y_{i-1j+1} - y_{i-1j-1})^2 - 12C^2}{h_{ij}^3}\right\} - 1 \\
&\quad - 2C\left\{\frac{4(h_{ij+1} + h_{ij-1}) + 6(y_{i-1j+1} - 2y_{i-1j} + y_{i-1j-1})}{h_{ij}^2}\right\} \\
&\quad - 48C\sum_{r=i+1}^{N}(1/h_{rj}) \\
S_{ij+1}^{ij} &= 2C\left\{\frac{4h_{ij} - h_{ij+1} + h_{ij-1}}{h_{ij}^2} + 3\sum_{r=i+1}^{N}\left(\frac{4h_{rj} - h_{rj+1} + h_{rj-1}}{h_{rj}^2}\right)\right\} \\
S_{kj-1}^{ij} &= 6C\left\{\frac{2h_{kj} + h_{kj+1} - h_{kj-1} - y_{k-1j+1} - y_{k-1j-1}}{h_{kj}^2}\right. \\
&\quad \left. + \sum_{r=k+1}^{N}\left(\frac{4h_{rj} + h_{rj+1} - h_{rj-1}}{h_{rj}^2}\right)\right\}
\end{aligned}\right\} \tag{13}$$

$$
\left.\begin{aligned}
S_{kj}^{ij} &= 6C\left\{\frac{(h_{k\,j+1}-h_{k\,j-1})^2+2(h_{k\,j+1}-h_{k\,j-1})(y_{k-1\,j+1}-y_{k-1\,j-1})}{h_{kj}^3}\right.\\
&\qquad \left.-\frac{2(h_{k\,j+1}+h_{k\,j-1})+4(y_{k-1\,j+1}-2y_{k-1\,j}+y_{k-1\,j-1})}{h_{kj}^2}-8\sum_{r=k+1}^{N}(1/h_{rj})\right\}-1\\
S_{kj+1}^{ij} &= -6C\left\{\frac{h_{k\,j+1}-h_{k\,j-1}-2h_{kj}+y_{k-1\,j+1}-y_{k-1\,j-1}}{h_{kj}^2}\right.\\
&\qquad \left.-\sum_{r=k+1}^{N}\left(\frac{h_{r\,j-1}+4h_{rj}-h_{r\,j+1}}{h_{rj}^2}\right)\right\}
\end{aligned}\right\}
$$

REFERENCES

1. R. W. Yeung, 'Numerical methods in free-surface flows', *Ann. Rev. Fluid Mech.*, **14**, 395 (1982).
2. R. V. Southwell and G. Vaisey, 'Relaxation methods applied to engineering problems XII. Fluid motions characterized by free streamlines. *Phil. Trans. Roy. Soc.*, **A240**, 117 (1946).
3. M. Ikegawa and K. Washizu, 'Finite Element analysis of flow over a spilway crest', *Int. J. Numer. Meth. Eng.*, **6**, 179 (1973).
4. E. Varoglu and W. D. L. Finn, 'Variable domain finite element analysis of free surface gravity flows', *Comp. and Fluids*, **6**, 103 (1978).
5. J. M. Aitchison, 'A finite element solution for critical flow over a weir', *Proc. 3rd Int. Conf. on Finite Elements in flow problems*, Banff, Alberta, Canada, 1980.
6. E. F. Toro, 'Finite element computation of free surface problems', *Ph.D. Thesis*, Mathematics Department, Teesside Polytechnic, England, 1982.
7. M. J. O'Carroll and E. F. Toro, 'Numerical computations of critical flow over a weir', *Int. J. Num. Meth. Fluids*, **4**, 499–509 (1984).
8. P. L. Betts and M. I. Assaat, 'Finite element computation of large-amplitude water waves', *Proc. 3rd Int. Conf. on Finite Elements in flow problems*, Banff, Alberta, Canada, 1980.
9. J. G. B. Byatt-Smith and M. S. Longuet-Higgins, 'On the speed and profile of steep solitary waves, *Proc. Roy. Soc. London*, **A350**, 175 (1976).
10. E. D. Cokelet, 'Steep gravity waves in water of arbitrary uniform depth', *Trans. Roy. Soc. London*, **A286**, 183 (1977).
11. L. W. Schwartz, 'Computer extension and analytic continuation of Stokes's expansion for gravity waves', *J. Fluid Mech.*, **62**, 553 (1974).
12. T. Y. Han and W. L. Chow, 'The study of sluicegate and sharp-crested weir through hodograph transformations', *Journal of Applied Mech., Trans. ASME*, **48**, 229 (1981).
13. L. V. Kantorovich and V. I. Krylov, *Approximate Methods of Higher Analysis*, P. Noodhoff Ltd., Groningen, The Netherlands, 1964.
14. M. J. O'Carroll and H. T. Harrison, 'Variational techniques for free-streamline problems', *Proc. 2nd Int. Symp. Finite Elements in Flow Problems*, Genoa, 1976, p. 485.
15. K. Rektorys, *The Method of Discretisation in Time and Partial Differential Equations*, D. Reidel Publishing Company, 1982.
16. K. M. Brown, 'Solutions of systems of non-linear equations. Algorithm 316', *Communications of the A.C.M.*, **10**, (11), (1967).