INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 151–173 (1996)

NUMERICAL COMPUTATION OF CRITICAL FLOW OVER A SPILLWAY BY MEANS OF MINIMIZATION OF VIOLATION OF NEGATIVE CURVATURE CONDITION

GEORGE MEJAK

Department of Mathematics and Mechanics, FNT, University of Ljubljana, Jadranska 19, Ljubljana, Slovenia

SUMMARY

A new method for the computation of two-dimensional critical flows over spillways is presented. The premise of the method is that at a critical value of the discharge coefficient that free boundary has no upstream waves. For the chosen spillway geometry without a toe section this requirement is equivalent to negative curvature of the free boundary. The method consists of two levels: at the inner level the corresponding free boundary value problem is solved for a fixed value of Q and at the outer level a critical value of Q is sought by minimization of the oscillation of the free boundary. An invaluable part of the method is the sensitivity anlaysis of the finite element stiffness matrix. The correctness of the numerical results is proved by scrutinizing the convergence rate by mesh refinement. Good agreement of the computed results with experimental data is achieved.

KEY WORDS: free surface flow; spillways; critical flow; finite element; sensitivity analysis; variable domain method

INTRODUCTION

Concomitant with the growth of computational facilities and the ready availability of powerful computers is the progress made in computational fluid mechanics. A significant aspect of this progress is that problems previously solved using relatively coarse discretizations can now be scrutinized using much finer discretizations. In this paper the numerical computation of the critical flow over a two-dimensional spillway is subjected to such scrutiny.

The computation of critical flow amounts to computing the steady flow over a spillway such that the approaching flow is subcritical. Since critical flow deals with the transition from subcritical upstream flow to supercritical downstream flow, the problem can be satisfactorily modelled within potential theory. The problem is classified as a free surface problem and is recognized as being rather difficult, because not only is the position of the free surface unknown in advance but also a value of the critical discharge coefficient has to be found as part of the solution. Akin to the problem of critical flow over a spillway is the determination of the critical flow over a weir or under a sluice gate. However, according to our computational experience and also according to other autors,[1,2] the spillway problem is more difficult and it was thus chosen as a representative example of critical flow.

Usually the numerical computation of the critical discharge coefficient is pursued via a two-level iteration algorithm. At the interior level the free boundary problem is solved for some assumed value of the discharge coefficient Q, while at the exterior level a new value of Q closer to the critical value is sought. The criterion of whether a certain value of Q is closer to the critical value is set in terms of the geometrical properties of the free boundary. In particular, it is required that the inlet depth of the free

CCC 0271–2091/96/030151–23

Received January 1994
Revised May 1995

boundary coincides with the subcritical depth. This requirement is usually supplemented with the condition that the free boundary has no waves upstream of the spillway crest. However, in our numerical computation we found that for a sufficiently long upstream region the inlet depth always coincides with the subcritical depth. The upstream region of spillway problems is usually rather long and thus in our numerical calculation the *no-wave* condition was chosen as the defining property of the critical discharge. In the case of a relatively short upstream region, as in References 3–5 where the critical flow over a weir is computed, the matching condition for the inlet depth gives some definite value of Q. However, what is lacking in this approach is a rigorous analysis of how the length of the upstream region affects the computation of the critical value of Q. We believe that such an analysis would reveal that the matching criterion is insufficient. The notion that the free boundary corresponding to the critical value of Q should have no waves has been recognized for quite a long time. At first[6,7] the notion was expressed vaguely that the critical value of Q is the value which gives the smoothest profile of the free boundary. Of course, this requirement is imprecise, since for a chosen discretization all computed free boundaries have the same degree of smoothness. More precise is Aitchison[8] with his phase-traking method. In this paper we introduce the more concise requirement that the upstream part of the free boundary has negative curvature. This restriction to the upstream part can be extended to the whole free boundary provided that the outlet is placed well before the toe where the curvature changes sign.

One of the most difficult aspects of the numerical calculation of critical flows by two-level algorithms is the fact that a solution to the inner iteration exists only for certain values of Q. According to our computational experience, for a relatively coarse discretization of about 30 free nodal points the convergence radius of the inner iteration with respect to Q is sufficiently large to begin with the empirical value of the critical discharge coefficient. With increasing refinement the convergence radius of Q shrinks and for a rather fine discretization of about 200 free nodal points the initial value of Q has to be given within an accuracy of 10^{-5}. An important observation is that for coarse discretizations the negative curvature condition in its discrete form is satisfied for quite a broad range of Q. With increasing refinement the above set of acceptable values of Q shrinks and becomes void. Therefore the critical value of Q for a discrete problem of flow over a spillway is actually defined as the value at which the negative curvature condition in its discrete form is least violated. In this way the outer iteration becomes a minimization algorithm with violation of the negative curvature condition as the objective function. The correctness of this approach is manifested by the fact that during mesh refinement it provides the inner iteration with convergent values of Q. The fact that for a coarse discretization multiple solutions exist explains why Finn and Varoglu's[9] method works within one iteration level. In their method an additional equatin resulting from the variation in kinetic energy with respect to Q is introduced into the inner free boundary iteration level and thus the position of the free boundary and the value of Q are computed simultaneously. However, we believe (see also the discussion in References 3 and 7) that their approach gives meaningful results only because of the coarse discretization, which as we now know allows quite a broad range of acceptable solutions.

STATEMENT OF THE PROBLEM

To allow comparison with previously published results[9,10] and experimental data,[11] a sharp-crested WES-standard spillway without piers was chosen. With reference to Figure 1, $AB =: \Gamma_{bed}$ is the flat bed of length L_1, BC $\cup$ CD $\cup$ DTE $=: \Gamma_{crest}$ is the spillway crest of length L_2, EF $=: \Gamma_{out}$ is the outlet, FG $=: \Gamma_{free}$ is the free boundary and GA $=: \Gamma_{in}$ is the inlet. The spillway crest consists of the vertical upstream face BC, two circular segments between C and D with radii r_1 and r_2 and a curve DT given in the local crest co-ordinate system (ξ, η) as $\xi^{1.85} = 2H_d^{0.85}\eta$, where H_d is the design head excluding the approach velocity head. The crest is prolonged by a straight line TE tangent to the curve DT. Here

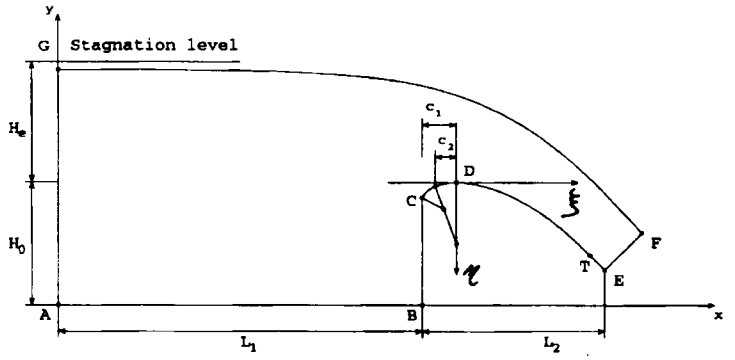

Figure 1. WES-standard spillway. Design head $H_d = 1{\cdot}8288$ m ($= 6$ ft), $L_1 = 3H_d$, $L_2 = 1{\cdot}615H_d$, $H_0 = H_d$, $r_1 = 0{\cdot}2H_d$, $r_2 = 0{\cdot}5$ H_d, $c_1 = 0{\cdot}282H_d$ and $c_2 = 0{\cdot}175H_d$

point T is chosen such that TE has in inclinatin of 45° in the local co-ordinate system (ξ, η). Introduction of the point of tangency is optional and is given here only to show some influence of the shape of the crest upon the flow.

The problem is modelled within potential theory. This means that the stagnation level is constant. In Figure 1 the stagnation level is given by $H_0 + H_e$, where H_0 is the height of the crest and H_e is the total energy level on the crest including the velocity of the approaching fluid. In this paper it is assumed that the stagnation level is given while the discharge coefficient Q is unknown. As is well known, the dual problem where Q is given and the stagnation level has to be found is very similar and can be analysed with almost the same approach. The formula which relates the dimensional flux Q_d per unit breadth and H_e is

$$Q_d = C_c H_e^{1.5}, \tag{1}$$

where C_c is the empirical discharge coefficient,[11] which in the particular case of $H_0 \geqslant 1{\cdot}33H_d$ equals 4·03.

Along the free boundary the Bernoulli equation

$$\tfrac{1}{2}V^2 + gY = g(H_e + H_0) \tag{2}$$

applies. Here V is the fluid speed along the boundary, Y is the free boundary elevation and g is the gravitational acceleration, which acts in the direction of the negative y-axis. Introducing dimensionless quantities using the mean x-component V_0 of the inlet velocity for the characteristic speed and H_d for the characteristic length, equation (2) is rewritten as

$$v^2 + \frac{2}{Fr}(y - h_e - h_0) = 0, \tag{3}$$

where capital letters in (2) are replaced by corresponding lowercase letters representing dimensionless quantities. Here $Fr = V_0^2/gH_d$ is the Froude number. Expressing the dimensionless flux Q per unit breadth as $Q = Q_d/\sqrt{(gH_d^3)}$ and noting that $Q_d = V_0 Y_1$, where Y_1 is the inlet depth, we have $Fr = Q^2/y_1^2$. In a numerical computation within a finite computational domain Ω the requirement that the approaching flow is subcritical is replaced by the conditions that the inlet depth y_1 coincides with the subcritical depth and that the upstream part of Γ_{free} does not have waves. To this end L_1 is taken sufficiently large and the assumption of a uniform x-component of the inlet velocity is made. In computation of the critical flow the location of the outlet is of less importance and thus Γ_{out} in our computations is placed quite arbitrarily at a distance $L_2 = 1{\cdot}615H_d$ downstream from the upstream face

BC of the crest. Since the velocity distribution along Γ_{out} is not known, it is assumed that Γ_{out} is the equipotential line. Note that this requires that Γ_{out} is orthogonal to both Γ_{free} and Γ_{crest}.

For a given value of Q the subcritical depth $S_2(Q)$ is defined as the largest root of the equation

$$Q^2 + 2S^3 - 2S^2(h_e + h_0) = 0, \tag{4}$$

which is just the Bernoulli equation of uniform flow over a flat bed with a given stagnation level $h_e + h_0$. Introducing (4) into the definition of the Froude number and using $y_1 = S_2(Q)$, it follows from (3) that

$$v^2 - \frac{y - h_e - h_0}{S_2(Q) - h_e - h_0} = 0. \tag{5}$$

To summarize, the continuum potential model of the critical flow over a spillway amounts to solving the following problem: find the velocity potential function φ, the positions of Γ_{free} and Γ_{out} and the discharge coefficient Q such that

$$\Gamma_{free} \textit{ has no upstream waves}, \tag{6}$$

$$\nabla^2\varphi = 0 \quad \text{in } \Omega, \tag{7}$$

$$\partial\varphi/\partial\mathbf{n} = 0 \quad \text{on } \Gamma_{bed} \cup \Gamma_{crest} \cup \Gamma_{free}, \tag{8}$$

$$\partial\varphi/\partial\mathbf{n} = -1 \quad \text{on } \Gamma_{in}, \tag{9}$$

$$\varphi = 0 \quad \text{on } \Gamma_{out}, \tag{10}$$

$$|\varphi|^2 - \frac{y - h_e - h_0}{S_2(Q) - h_e - h_0} = 0 \quad \text{on } \Gamma_{free}, \tag{11}$$

$$y_G - S_2(Q) = 0. \tag{12}$$

Here the subscript G denotes that y_G is the y-co-ordinate of the inlet point G. Note that conditions (8) and (10) imply that Γ_{out} is orthogonal to both Γ_{free} and Γ_{crest} and thus the position of Γ_{out}, owing to the unknown position of Γ_{free}, is also not known in advance. Usually Γ_{out} is approximated by a straight line orthogonal to Γ_{crest}. Although this simplifies the code significantly, we require in this paper that Γ_{out} is orthogonal to Γ_{free}, since this makes an important difference in the minimization of the violation of the negative curvature condition. Pausing for a moment, we note that the discretization of problem (6)–(12) results in an ill-conditioned problem. Indeed, for a sufficiently long upstream region the *no-wave* condition (6) implies that the inlet Γ_{in} is nearly orthogonal to Γ_{free}. This together with (9) implies that at the inlet point G, $v \approx 1$ and thus (11) and (12) are almost identical equations. Therefore in the numerical calculation equation (12) is simply omitted. The above argument shows that the numerical solution of (6)–(11) also satisfies (12) to high accuracy.

DISCRETIZATION OF THE PROBLEM

The problem is discretized using the finite element method. Since (11) involves values of the gradient of φ, the isoparameteric Hermite–Zienkiewicz element is chosen. Let us denote by Ω_h the finite element discretization of Ω and by $\mathbf{u} \in \mathbb{R}^r$ a vector of degrees of freedom corresponding to the finite element discretization φ_h of φ. Since from now on all discussion refers to Ω_h, the same letters as for Ω are used for the characteristic points and boundary segments of Ω_h. The finite element mesh is determined by the conformal mapping from the reference domain Ω_r onto Ω_h. For Ω_r is chosen a spillway shape domain where the free boundary is replaced by a spline consisting of two segments. The first segment is a straight line parallel to the x-axis and extends from the inlet point G to point D′

placed above the highest point of the crest, while the second segment is the interpolating parabola through points D′ and F with a smooth fit at D′. Points F and G are determined by using the empirical value of Q and assuming that the flow is uniform and subcritical/supercritical at the inlet/outlet. Here the outlet is just a straight line orthogonal to the crest at point E. Actually, to get a better approximation of Γ_{free}, point D′ is lowered slightly. In this way the determined boundary FG of Ω_r also serves as an initial approximation of the free boundary of Ω_h. The conformal mapping from Ω_r onto Ω_h is approximated by using linear triangular elements. The finite element stiffness matrix of this approximation is of size approximately one-ninth of the flow finite stiffness matrix and needs to be computed only at the initialization step. The reference finite element mesh on Ω_r is determined by five parameters n_i, $i = 1, \ldots, 5$; see Figure 2. Here n_1 is the number of y-layers above the crest, $n_1 + n_2$ is the number of y-layers of the upstream region, n_3 is the number of x-layers of the upstream region, n_4 is the number of x-layers of the crest between C and D and n_5 is the number of x-layers of the crest between D and E. Note in Figure 2 that the layers along Γ_{free} and Γ_{crest} are further refined into sublayers. Mesh refinement is also carried out in the vicinity of point C. The spacing between the x-co-ordinates of the finite element nodes in the upstream region is uniform, while the mesh spacing in the y-direction between B and C behaves as $(y_C - y)^{1\cdot4}$. Here y_C is the y-co-ordinate of point C. A complete list of all meshes used in the paper is given in Table I, there N_e is the number of finite elements and s is the number of finite element nodes along Γ_{free}.

As already noted, Γ_{out} is required to be orthogonal to both Γ_{free} and Γ_{crest}. The simplest way to achieve this is to approximate Γ_{out} by a parabola. This approximation is uniquely determined by interpolation through point F and the orthogonality condition between Γ_{free} and Γ_{crest}. Note that this approximation also affects the position of point E. The co-ordinates of the finite element nodes along Γ_{free} are not known in advance and are thus subjected to certain displacements during the free boundary iteration. Displacements of all upstream free boundary nodes up to point D′ are allowed only in the y-direction, while the free boundary nodes along the upper nappe are variable in the directions of the normals to the lower nappe. In this way, with each free boundary node is associated one unknown parameter and together they constitute the parameter set of the free boundary problem. Using the isoparametric Hermite–Zienkiewicz element, Γ_{free} is specified not only by the positions of the boundary nodes but also by the directional cosines of the tangential derivatives of the free boundary.[12,13] Although it is possible to include directional cosines in the parameter set, we proceed with the simplification that directional cosines are determined by interpolation through the free boundary nodes. As we shall see, it is preferable to have a local interpolation in the sense that the tangential derivatives at the finite element node P_i is not affected by the positions of nodes P_j far away

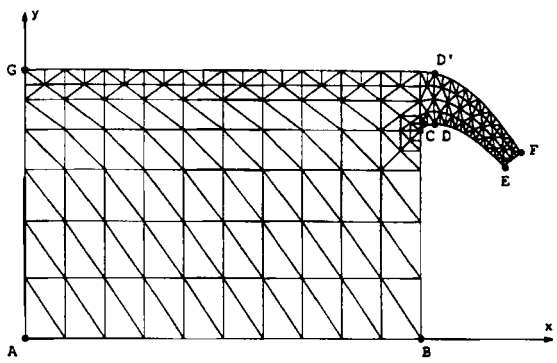

Figure 2. Reference spillway shape domain Ω_r. Mesh parameters $n_1 = 2$, $n_2 = 4$, $n_3 = 10$, $n_4 = 1$ and $n_5 = 7$. The mesh consists of 302 elements and has 37 boundary nodes along Γ_{free}

Table I. List of mesh definitions. A mesh with parameters n_1, n_2, n_3, n_4 and n_5 is referred to as mesh $M_{n_1 n_2 n_3 n_4 n_5}$

Mesh	n_1	n_2	n_3	n_4	n_5	N_e	s
$M_{2405105}$	2	4	5	1	5	189	23
$M_{2410107}$	2	4	10	1	7	302	37
$M_{2420110}$	2	4	20	1	0	514	63
$M_{2430110}$	2	4	30	1	10	684	83
$M_{2430210}$	2	4	30	2	10	698	85
$M_{2830210}$	2	8	30	2	10	935	85
$M_{4830210}$	4	8	30	2	10	1106	85
$M_{4830220}$	4	8	30	2	20	1286	105
$M_{4840220}$	4	8	40	2	20	1576	125
$M_{6840220}$	6	8	40	2	20	1824	125
$M_{8840220}$	8	8	40	2	20	2072	125
$M_{8840320}$	8	8	40	3	20	2098	127
$M_{6860320}$	6	8	60	3	20	2506	167
$M_{6880320}$	6	8	80	3	20	3166	207
$M_{7880320}$	7	8	80	3	20	3372	207
$M_{8880320}$	8	8	80	3	20	3578	207
$M_{9880320}$	9	8	80	3	20	3784	207
$M_{10880320}$	10	8	80	3	20	3990	207
$M_{10880420}$	10	8	80	4	20	4029	209
$M_{101280420}$	10	12	80	4	20	4660	209

from P_i. To this end the tangential derivaties at $P_i = (x_i, y_i)$ is determined by differentiation of x and y polynomial interpolation through nodes $\{x_{i-1}, x_i, x_{i+1}\}$ and $\{y_{i-1}, y_i, y_{i+1}\}$ respectively. Here the polygonal arc length between nodes P_{i-1}, P_i and P_{i+1} is chosen for the independent variable. At the end point F the interpolation is through nodes P_{s-2}, P_{s-1} and $P_s = F$ and hence, owing to the orthogonality condition between Γ_{free} and Γ_{out}, the position of Γ_{out} depends upon the variation in these nodes. The x-co-ordinates of the nodes along Γ_{out} are determined by the affine mapping between the image of the projection of the outlet boundary of Ω_r onto the x-axis and the image of the projection of Γ_{out} onto the x-axis of the computational domain Ω_h. The inlet point G is common to Γ_{free} and Γ_{in} and is thus variable. A simple approach wherein along Γ_{in} only G is variable and all other nodes are fixed may, in the case of a very fine discretization relative to the displacement of G, result in too distorted elements. Therefore all nodes along Γ_{in} are variable and are moved by scaling their y-co-ordinates by the displacement of G. The finite element nodes along Γ_{bed} and Γ_{crest}, except the end point E, are fixed and have the same co-ordinates as their corresponding nodes on the boundary of Ω_r. In this way the boundary data of the conformal mapping from Ω_r onto Ω_h are uniquely determined by Γ_{free} and thus the positions of the finite element nodes of Ω_h depend only upon the co-ordinates of the nodes along Γ_{free}. This property is very important for the free boundary iteration since it eliminates the dependence of the solution upon the way the iteration evolves.

By the above approximation Γ_{free} is a piecewise smooth curve with global regularity of class C^1. At the free nodal points the curvature is not defined and thus it is natural to replace the negative curvature condition by the requirement that the upstream part of Γ_{free} is a concave curve. Noting that the positions of the free boundary nodes are of primal interest, the concavity condition is further relaxed to the requirement that

$$(\overrightarrow{P_i P_{i+1}} \times \overrightarrow{P_{i-1} P_i}) \times \vec{k} \geqslant 0. \tag{13}$$

Here the subscript i runs from the inlet to the outler and $\vec{k}$ is the unit positively oriented vector orthogonal to the x- and y-axes. Since our analysis does not include the toe section, the index i is allowed to run from $i=2$ to $s-1$, where s is the number of free boundary nodes. Accumulating changes of direction of $\overrightarrow{P_iP_{i+1}}$, we define

$$\Phi_0(\Gamma_{\text{free}}) = \cos^{-1}\left(\frac{\overrightarrow{P_1P_2}\cdot\vec{i}}{|\overrightarrow{P_1P_2}|}\right) + \sum_{i=2}^{s-1} \cos^{-1}\left(\frac{\overrightarrow{P_{i-1}P_i}\cdot\overrightarrow{P_iP_{i+1}}}{|\overrightarrow{P_{i-1}P_i}||\overrightarrow{P_iP_{i+1}}|}\right), \tag{14}$$

where $\vec{i}$ is the unit basis vector of the x-axis. Note that $\Phi_0(\Gamma_{\text{free}})$ measures the oscillation of Γ_{free}. Note also that the sum in (14) is just a discrete form of

$$\int_G^F |\kappa| d\sigma, \tag{15}$$

where κ is the curvature and $d\sigma$ is the arc length element. For a sufficiently long upstream region it is clear the $y_1 \geqslant y_2$ and thus, if (13) is satisfied for all $i \in \{2, \ldots, s-1\}$, it follows that for

$$\Phi(\Gamma_{\text{free}}) = \Phi_0(\Gamma_{\text{free}}) - \cos^{-1}\left(\frac{\overrightarrow{P_{s-1}P_s}\cdot\vec{i}}{|\overrightarrow{P_{s-1}P_s}|}\right) \tag{16}$$

we have $\Phi(\Gamma_{\text{free}})=0$. Moreover, any violation of (13) obviously results in $\Phi(\Gamma_{\text{free}})>0$ and therefore $\Phi(\Gamma_{\text{free}})$ measures the violation of the discrete negative condition. Since for certain mesh discretizations Γ_{free} is not a concave curve, $\Phi(\Gamma_{\text{free}})=0$ is too stiff a condition to be introduced directly into the numerical calculation and thus instead the minimization of $\Phi(\Gamma_{\text{free}})$ is attempted. To summarize, with the continuum problem (6)–(12) is associated the following optimization problem: for a given finite element discretization of Ω_h and mapping from Ω_r onto Ω_h find Q_c such that

$$\Phi(\Gamma_{\text{free}}(Q_c)) = \min_Q\{\Phi(\Gamma_{\text{free}}(Q))\}, \tag{17}$$

where $\Gamma_{\text{free}}(Q)$ is the free boundary of the discrete free boundary problem (7)–(11) with a fixed value Q of the discharge coefficient. In the following, problem (17) is referred to as the Q-problem.

An alternative to minimization of $\Phi(\Gamma_{\text{free}})$ is the minimization of $\Phi_0(\Gamma_{\text{free}})$. There is no significant difference between the two alternatives. However, for concave $\Gamma_{\text{free}}(Q_c)$ the target value of $\Phi(\Gamma_{\text{free}})$ is known and thus $\Phi(\Gamma_{\text{free}})$ is preferred for the objective function.

SOLUTION OF THE DISCRETE FREE BOUNDARY PROBLEM

In this section the discrete solution to the free boundary problem (7)–(11) for a fixed value of Q is sought. Since a solution exists only for certain values of Q, the solution algorithm has to be reliable with a well-tested convergence behaviour. Our previous[4,5] computational experience with the variable domain method was encouraging and thus this method is implemented in the present paper.

Let us denote by $\mathbf{h} \in \mathbb{R}^s$ a vector corresponding to the parameter set. The co-ordinates of $\mathbf{h}$ are denoted by h_i and are ordered in such a way that h_i runs from the inlet to the outlet. With this notation the dependence of Ω_h upon Γ_{free} is denoted by $\Omega(\mathbf{h})$ and the free boundary problem (7)–(11) is referred to as the $\mathbf{h}$-problem. For a given domain $\Omega(\mathbf{h})$ a discretization of (7)–(10) is just a discretization of the flow problem. Denoting by $\mathbf{K}(\mathbf{h})$ the finite element stiffness matrix and by $\mathbf{b}(h_1)$ the load vector, the linear system $\mathbf{K}(\mathbf{h})\mathbf{u}=\mathbf{b}(h_1)$ is obtained for the unknown vector $\mathbf{u}$. Here the notations $\mathbf{K}(\mathbf{h})$ and $\mathbf{b}(h_1)$ indicate that $\mathbf{K}(\mathbf{h})$ and $\mathbf{b}(h_1)$ depend upon $\mathbf{h}$ and h_1. The dependence of $\mathbf{K}(\mathbf{h})$ is obvious, while the dependence of the load vector follows from the variable position of the finite element nodes along Γ_{in}.

It should be noted that $\mathbf{b}(h_1)$ does not depend upon Γ_{out}. Discretization of the Bernoulli condition (11) can be performed in several ways. Since the Hermite–Zienkiewicz element is implemented, the simplest way to impose (11) is by collocation at the finite element nodes along Γ_{free}. However, it was found that in this case the sensitivity of the **h**-problem to the accuracy of the value of Q is very high. For values $Q \neq Q_c$ this sensitivity is manifested by oscillation of the upstream part of the free boundary. Similar oscillations of Γ_{free} were observed and reported in Reference 2 as the uncertain region. The wavelength of the oscillation is proportional to $2L_1/n_3$ while the amplitude is proportional to $|Q - Q_c|/n_1$. When such oscillation occurs and the amplitude exceeds a critical value, the variable method is prone to failure. However, we report that in the computation of the critical flow over a weir[4,5] with a short upstream region such oscillations are not present and hence the collocation approach is perfectly acceptable.

In this paper the Bernoulli condition (11) is discretized by using the weighted residual method. We note that in the case of the weighted residual method the sensitivity of the **h**-problem to the accuracy of Q is of lesser degree. Denoting by $\{w_j: j = 1, 2, \ldots, s\}$ a suitably chosen collection of weighting functions, equation (11) is discretized as

$$\int_{\Gamma_{\text{free}}} \left(|\nabla \varphi_h|^2 - \frac{y - h_e - h_0}{S_2(Q) - h_e - h_0} \right) w_j \, d\sigma = 0, \tag{18}$$

$j \in \{1, 2, \ldots, s\}$. For the weighting functions, linear shape functions are chosen. In particular, w_j is a piecewise linear interpolating function of the arc length such that w_j has support between nodes P_{j-1} and P_{j+1} and such that $w_j(P_k) = \delta_{jk}$. Denoting by $\mathbf{c} \in \mathbb{R}^s$ a vector whose jth component equals (18), discretization of (11) results in a non-linear system $\mathbf{c}(\mathbf{u}, \mathbf{h}) = \mathbf{0}$. Here the notation indicates that **c** depends non-linearly on **u** and **h**. Summing up, discretization of the free boundary value problem (7)–(11) requires us to solve the non-linear system

$$\begin{bmatrix} \mathbf{K}(\mathbf{h})\mathbf{u} - \mathbf{b}(h_1) \\ \mathbf{c}(\mathbf{u}, \mathbf{h}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{19}$$

Noting that **u** appears linearly in the first set of equations (19), **u** is eliminated and thus a solution to the non-linear system

$$\mathbf{f}(\mathbf{h}) = \mathbf{c}(\mathbf{K}^{-1}(\mathbf{h})\mathbf{b}(h_1), \mathbf{h}) = \mathbf{0} \tag{20}$$

is sought. The solution is attempted by the Newton–Raphson method. Denoting by $\mathbf{h}^{(k)}$ the vector **h** at the kth iteration step, $\mathbf{h}^{(k+1)}$ is determined by

$$\mathbf{h}^{(k+1)} = \mathbf{h}^{(k)} + \Delta\mathbf{h}^{(k)} = \mathbf{h}^{(k)} - [D\mathbf{f}(\mathbf{h}^{(k)})]^{-1}\mathbf{f}(\mathbf{h}^{(k)}), \tag{21}$$

where $\mathrm{D}\mathbf{f}(\mathbf{h}^{(k)})$ is the Jacobian of **f** computed at $\mathbf{h}^{(k)}$. The main computational task lies in the computation of the Jacobian. Using the chain rule, it follows that

$$\begin{aligned} D\mathbf{f}(\mathbf{h}) &= \frac{\partial \mathbf{c}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{h}} + \frac{\partial \mathbf{c}}{\partial \mathbf{h}} \\ &= \frac{\partial \mathbf{c}}{\partial \mathbf{u}} \left(-\mathbf{K}^{-1}(\mathbf{h}) \frac{\partial \mathbf{K}(\mathbf{h})}{\partial \mathbf{h}} \mathbf{K}^{-1}(\mathbf{h})\mathbf{b}(h_1) + \mathbf{K}^{-1}(\mathbf{h}) \frac{\partial \mathbf{b}(h_1)}{\partial \mathbf{h}} \right) + \frac{\partial \mathbf{c}}{\partial \mathbf{h}}, \end{aligned} \tag{22}$$

where $\partial \mathbf{c}/\partial \mathbf{u} \in \mathscr{L}(\mathbb{R}^r, \mathbb{R}^s) \approx \mathbb{R}^{s \times r}$, $\partial \mathbf{u}/\partial \mathbf{h} \in \mathbb{R}^{r \times s}$, $\partial \mathbf{c}/\partial \mathbf{h} \in \mathbb{R}^{s \times s}$ and $\partial \mathbf{K}(\mathbf{h})/\partial \mathbf{h} \in \mathscr{L}(\mathbb{R}^s, \mathbb{R}^{r \times r})$. Here $\mathscr{L}(X, Y)$ denotes the space of all linear transformations from a vector space X to a vector space Y.

The computation of the Jacobian is organized as follows.

Step 1. Perform factorization of $\mathbf{K}(\mathbf{h})$ and solve $\mathbf{K}(\mathbf{h})\mathbf{u} = \mathbf{b}(h_1)$.
Step 2. Compute $\partial\mathbf{c}/\partial\mathbf{u}$, $\partial\mathbf{c}/\partial\mathbf{h}$ and store $\partial\mathbf{c}/\partial\mathbf{h}$ in $\mathbf{Df}(\mathbf{h})$. For each $i \in \{1, \ldots, s\}$ do:
Step 3.i. Compute $-\partial\mathbf{K}(\mathbf{h})/\partial h_i$, multiply the result by $\mathbf{u}$ and for $i = 1$ add to this result $\partial\mathbf{b}(\mathbf{h})/\partial h_1$. Store the result in $\mathbf{q}_i$.
Step 4.*i.* Solve $\mathbf{K}(\mathbf{h})\mathbf{p}_i = \mathbf{q}_i$ and add the result to the ith column of $\mathbf{Df}(\mathbf{h})$.

We now discuss how to compute individual parts of the Jacobian. Since c_j depends only upon those degrees of freedom which are associated with elements which have nodes along the support of w_j, $\partial\mathbf{c}/\partial\mathbf{u}$ is a rather sparse matrix. Components of $\mathbf{c}$ are readily expressed as the combination of terms

$$c_{jk} = \int_{P_j}^{P_{j+1}} \left(|\nabla\varphi_h|^2 - \frac{y - h_e - h_0}{S_2(Q) - h_e - h_0} \right) \hat{M}_k \, d\sigma, \quad k \in \{1, 2\}, \tag{23}$$

where $\hat{M}_k$ are the linear finite basis functions of the arc P_jP_{j+1}. Therefore it suffices to consider only the computations of $\partial c_{jk}/\partial\mathbf{u}$. Let us denote by Ω_e the element with its side along P_jP_{j+1}. With this element is associated a vector of the local finite element basis functions, $\mathbf{N} = [N_I]_{I=1,\ldots,9}$, and a vector of the local degrees of freedom, $\mathbf{U}_e = [U_I]_{I=1,\ldots,9}$. Since the Hermite–Zienkiewicz element is used, the local degrees of freedom are not the same as the global ones but are instead expressed by the relation[13] $U_I = H_{IJ} u_{e(J)}$. Here and also in the sequel the summation convection over repeated indices applies; e: $\{1, \ldots, 9\} \to \mathbb{N}$ is the transformation from the local enumeration into the global one and H_{IJ} are the elements of the matrix

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_3 \end{bmatrix} \in \mathbb{R}^{9\times 9}, \tag{24}$$

where $0 \in \mathbb{R}^{3\times 3}$ and

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \xi_{1,3k-1} & \xi_{2,3k-1} \\ 0 & \xi_{1,3k} & \xi_{2,3k} \end{bmatrix} \in \mathbb{R}^{3\times 3}, \quad k = 1, 2, 3. \tag{25}$$

Here $\Xi = [\xi_{mJ}] \in \mathbb{R}^{2\times 9}$ is the matrix of the parametrs of the isoparametric transformation from the master Hermite–Zienkiewicz element onto Ω_e. The uppercase indices refer to the local enumeration of the nodes and are arranged in such a way that the global degrees of freedom with indices $e(3k-2)$, $e(3k-1)$ and $e(3k)$, $k = 1, 2, 3$, correspond to the function value and values of x- and y-partial derivatives. The subscript $m \in \{1, 2\}$ refers to the x- and y-co-ordinates of the ordered pair $\xi_{\bullet,N}$. Note that $\xi_{\bullet,3k-2}$, $k = 1, 2, 3$, are co-ordinates of the vertices of Ω_e. It follows now that for the global index $l = e(L)$ we have

$$\frac{\partial}{\partial u_l} \nabla\varphi_h = \frac{\partial}{\partial u_l} \nabla N_K U_K = \frac{\partial}{\partial u_{e(L)}} \nabla N_K H_{KL} u_{e(L)} = \nabla N_K H_{KL} \tag{26}$$

and thus

$$\frac{\partial}{\partial u_l} c_{jk} = 2 \int_{P_j}^{P_{j+1}} \nabla\varphi_h \nabla N_K H_{KL} \hat{M}_k \, d\sigma. \tag{27}$$

Owing to the local dependence of c_{jk} upon h_i, it suffices to discuss only the computation of $\partial c_{jk}/\partial h_i$ for $i \in \{\max\{j-1, 1\}, j, j+1, \min\{s, j+2\}\}$. Note that the bandwidth of $\partial\mathbf{c}/\partial\mathbf{h}$ is five. Using the chain rule, we have

$$\frac{\partial c_{jk}}{\partial h_i} = \frac{\partial c_{jk}}{\partial \xi_{mN}} \frac{\partial \xi_{mN}}{\partial h_i}. \tag{28}$$

The last term on the right is given by the known dependence of the parameters of the isoparametric upon h_i. In the computation of $\partial c_{jk}/\partial \xi_{mN}$ the partial derivatives $\partial y/\partial \xi_{mN}$, $\partial |\nabla \varphi_h|^2/\partial \xi_{mN}$ and $\partial d\sigma/\partial \xi_{mN}$ are encountered. The computation of $\partial y/\partial \xi_{mN}$ is trivial, since $y = \xi_{2N} N_N$ and thus $\partial y/\partial \xi_{mN} = \delta_{m2} N_N$. To compute $\partial |\nabla \varphi_h|^2/\partial \xi_{mN}$, we use the formula[14]

$$\frac{\partial}{\partial \xi_{mN}} N_{K;l} = -N_{K;m} N_{N;l}, \tag{29}$$

where the subscripts l, $m \in \{1, 2\}$ after the semicolon denote partial differentiation with respect to x and y. Using (29), it follows that

$$\begin{aligned}\frac{\partial |\nabla \varphi_h|^2}{\partial \xi_{mN}} &= 2\varphi_{h,l} \frac{\partial}{\partial \xi_{mN}} \varphi_{h,l} = 2\varphi_{h,l} \frac{\partial}{\partial \xi_{mN}} N_{K;l} U_K \\ &= 2\varphi_{h,l} \left(\frac{\partial N_{K;l}}{\partial \xi_{mN}} U_K + N_{K;l} \frac{\partial H_{KL}}{\partial \xi_{mN}} u_{e(L)} \right) \\ &= 2\varphi_{h,l} \left(-N_{K;m} N_{N;l} H_{KL} + N_{K;l} \frac{\partial H_{KL}}{\partial \xi_{mN}} \right) u_{e(L)},\end{aligned}$$

where $\partial H_{KL}/\partial \xi_{mN} \in \{0, 1\}$ is readily determined using (24) and (25). In the computation of $\partial d\sigma/\partial \xi_{mN}$ we first note that the element side between P_j and P_{j+1} is given by

$$x_m(t) = \xi_{mN} \hat{N}_N(\hat{x}_1(t), \hat{x}_2(t)), \quad t \in [0, 1], \tag{31}$$

where $\hat{N}_n$ are the master element basis functions and $\hat{x}_m(t)$ are the side parameter functions. For example, in the case where the side $P_j P_{j+1}$ is the image of the master element side $\hat{P}_3 \hat{P}_1$, we have $\hat{x}_1(t) = 0$ and $\hat{x}_2(t) = 1 - t$. The arc length element is $d\sigma = \sqrt{(\dot{x}_1^2 + \dot{x}_2^2)} dt$ and

$$\frac{\partial d\sigma}{\partial \xi_{mN}} = \frac{1}{d\sigma} \dot{x}_m \hat{N}_{N;l} \frac{d\hat{x}_l}{dt}. \tag{32}$$

The partial differentiation of the load vector is computed element by element. Let us denote by Ω_e one of the elements which has a side along Γ_{in} and by $\mathbf{b}_e \in \mathbb{R}^9$ the contribution of Ω_e to $\mathbf{b}(h_1)$. Applying the same technique as above, the computation of $\partial \mathbf{b}_e/\partial h_1$ follows by noting that $\mathbf{b}_e = \mathbf{H}^T \hat{\mathbf{b}}$, where $\hat{\mathbf{b}} \in \mathbb{R}^9$ has components

$$\hat{b}_I = -\int_0^1 \hat{N}_I(x_1(t), x_2(t)) d\sigma. \tag{33}$$

However, using the fact that the side of Ω_e along Γ_{in} is a straight line in the direction of the y-axis, it follows that

$$\frac{\partial b_{eI}}{\partial h_1} = \frac{\delta(I)}{y_1} b_{eI}, \tag{34}$$

where y_1 is the y-co-ordinate of point G and $\delta(I) \in \{0, 1, 2\}$. Note that in the derivation of (34) the scaling relation between the displacement of point G and the y-co-ordinates of the nodes along Γ_{in} is used. The particular form of $\delta(I)$ depends upon the enumeration of the nodes of Ω_e. For example, if in the local enumeration of Ω_e the vertices $\hat{P}_3$ and $\hat{P}_1$ lie along Γ_{in}, then $\delta(2) = \delta(4) = \delta(5) = \delta(6) = \delta(8) = 0$, $\delta(1) = \delta(7) = 1$ and $\delta(3) = \delta(9) = 2$. Formula (34) of course also follows directly from (33) by noting that only the length of the element side depends upon h_1.

We now turn to the computation of $\partial \mathbf{K}(\mathbf{h})/\partial h_i$. It is clear that $\mathbf{K}(\mathbf{h})$ depends upon $\mathbf{h}$ owing to the variation in the finite element nodes. However, since the Hermite–Zienkiewicz element is used and the

boundary segment Γ_{out} where the Dirichlet boundary contion is imposed is variable, $\mathbf{K}(\mathbf{h})$ also depends upon the variation in the boundary conditions. Indeed, for each node along Γ_{out} a constraint

$$\frac{\partial \varphi_h}{\partial x} t_1 + \frac{\partial \varphi_h}{\partial y} t_2 = 0 \tag{35}$$

is introduced by the Lagrange multiplier method[15] into the finite element stiffness matrix. Here $\vec{t} = (t_1, t_2)$ is the tangent to Γ_{out} at the finite element node. Since $\vec{t}$ depends upon Γ_{out} and thus upon h_{s-2}, h_{s-1} and h_s, the variation in the constraints (35) with respect to $\mathbf{h}$ has to be included in the computation of $\partial \mathbf{K}(\mathbf{h})/\partial h_i$. For a known dependence of Γ_{out} upon Γ_{free} this can be done in a straightforward way.

The computation of $\partial \mathbf{K}(\mathbf{h})/\partial h_i$, owing to the variation in the mesh, is performed element by element. The contribution of the element Ω_e to $\mathbf{K}(\mathbf{h})$ is $\mathbf{H}_e^T \mathbf{K}_e \mathbf{H}_e$, where $\mathbf{K}_e \in \mathbb{R}^{9\times 9}$ has elements

$$K_{e,IJ} = \int_{\Omega_e} N_{I;l} N_{J;l} \mathrm{d}\Omega_e, \tag{36}$$

and thus the contribution of Ω_e to $\partial \mathbf{K}(\mathbf{h})/\partial h_i$ is

$$\frac{\partial}{\partial h_i}(\mathbf{H}_e^T \mathbf{K}_e \mathbf{H}_e) = \left[\frac{\partial \mathbf{H}_e^T}{\partial \xi_{mN}} \mathbf{K}_e \mathbf{H}_e + \left(\frac{\partial \mathbf{H}_e^T}{\partial \xi_{mN}} \mathbf{K}_e \mathbf{H}_e \right)^T + \mathbf{H}_e^T \frac{\partial \mathbf{K}_e}{\partial \xi_{mN}} \mathbf{H}_e \right] \frac{\partial \xi_{mN}}{\partial h_i}. \tag{37}$$

A term in (37) not yet encountered is $\partial \mathbf{K}_e/\partial \xi_{mN}$, which is computed using (29) and the known fact that the partial differentiation of the volume element $\mathrm{d}\Omega_e$ is given by[14]

$$\frac{\partial \mathrm{d}\Omega_e}{\partial \xi_{mN}} = \mathrm{d}\Omega_e N_{N;m}. \tag{38}$$

Therefore, in order to compute $\partial \mathbf{K}(\mathbf{h})/\partial h_i$, the partial derivatives $\mathbf{H}_e^T(\partial \mathbf{K}_e/\partial \xi_{mN})\mathbf{H}_e$ for all elements Ω_e have to be known. Actually, owing to the multiplication of $\partial \mathbf{K}(\mathbf{h})/\partial h_i$ by $\mathbf{u}$, it suffices to know $\mathbf{H}_e^T(\partial \mathbf{K}_e/\partial \xi_{mN})\mathbf{U}_e$. Since for each $i \in \{1, \ldots, s\}$ all these element contributions to $[\partial \mathbf{K}(\mathbf{h})/\partial h_i]\mathbf{u}$ have to be known simultaneously, all $\mathbf{H}_e^T(\partial \mathbf{K}_e \partial \xi_{mN})\mathbf{U}_e$ are computed immediately after Step 1 and are stored as data of size $18 \times 9 \times N_e$ floating point numbers. Here N_e is the number of finite elements of Ω_h. The computational cost of $\mathbf{H}_e^T(\partial \mathbf{K}\mathbf{e}/\partial \xi_{mN})\mathbf{U}_e$ is slightly higher than the cost of assembling $\mathbf{K}_e$ and thus the cost of computing all required element derivatives is approximately of the same order as the cost of assembling the stiffness matrix.

Estimating the computational cost, we note that the total cost of Steps 4.i, $i = 1, \ldots, s$, is rather high. Indeed, denoting by d the average half-bandwidth of $\mathbf{K}(\mathbf{h})$, the computational cost of Step 4 is $O(rds)$ while the cost of Step 1 is $O(rd^2 + rd)$. Using the skyline storage schema and y-directed enumeration of the nodes, a typical ratio $s{:}d$ is between two and four. For example, for the mesh with parameters $n_1 = 10$, $n_2 = 8$, $n_3 = 80$, $n_4 = 3$ and $n_5 = 20$ we have $N_e = 3990$, $r = 6549$, $s = 207$ and $d = 58$ and thus the cost of Step 4 is higher by a factor of approximately 3·6 than that of Step 1.

Taking into account also the cost of computing $\partial \xi_{mN}/\partial h_i$, estimation of the cost of computation of the Jacobian $\mathrm{D}\mathbf{f}(\mathbf{h})$ shows that the cost of each Newton–Raphson iteration step is comparable with the cost of solving four flow problems on a fixed domain. Therefore, in order to avoid expensive computations of the Jacobian, it seems natural to try Broyden's method[16,17] instead of the Newton–Raphson method. Using Broyden's method, the problem remains of how to get the initial approximation to the Jacobian. Sometimes Broyden's method works if it begins with the appropriate scaled identity matrix. However, according to our numerical experience, this approach requires a fairly good initial guess of the free boundary and despite this quite often results in very slow convergence. To overcome these difficulties, the initial approximation to the Jacobian is computed as if only finite

elements along $\Gamma_{in} \cup \Gamma_{free} \cup \Gamma_{out}$ are variable and all other interior elements are fixed. Although this simplification reduces the computational cost by about 25% and also reduces the storage requirement, it does not notably simplify the code. In the following we refer to the method where such an approximation to the Jacobian is applied as the reduced Newton–Raphson (RNR) method. Only a further simplification where in the computation of $\partial \mathbf{K}(\mathbf{h})/\partial h_i$ isoparameteric elements are replaced by rectilinear ones and where partial differentiation with respect to h_i is approximated by a finite-difference formula will notably simplify the code. However, this further simplification, which is for certain problems quite acceptable, does not work satisfactorily in the present situation. The reason is that in the case of oscillation of Γ_{free} the variation in the stiffness matrix is not adequately modelled by rectilinear elements. The computation of $\partial \xi_{mN}/\partial h_i$ also requires differentiation of the directional cosines of the tangents to $\Gamma_{free} \cup \Gamma_{out}$. Although this can be done analytically, we instead, in accordance with the loss of accuracy due to the simplification that only elements along the movable boundary are variable, chose the finite difference approximation with a step size of order $\sqrt{\epsilon_m}$, where ε_m is machine epsilon.

It is well known that a switch from the Newton–Raphson method to Broyden's method after the first iteration step is often premature. Therefore the switch is made only after the length of the RNR method step, $\Delta \mathbf{h}$, falls below a certain threshold. Of course, not all components of $\Delta \mathbf{h}$ are of the same magnitude. Usually the magnitude of quite a large number of the components is relatively small well before the length of the step falls below the threshold. Now, since Steps 3.*i* and 4.*i* can be computed for individual indices *i*, it seems natural to compute $\mathbf{p}_i^{(k)} = -\mathbf{K}^{-1}(\mathbf{h}^{(k)})[\partial \mathbf{K}(\mathbf{h}^{(k)})/\partial h_i]\mathbf{u}^{(k)}$ at the *k*th iteration step only for those indices *i* for which the *i*th component of $\Delta \mathbf{h}^{(k-1)}$ is not small enough and to set $\mathbf{p}_i^{(k)} = \mathbf{p}_i^{(k-1)}$ for all other indices *i*. Since $\mathbf{p}_i^{(k)}$ is multiplied from the left by $\partial \mathbf{c}/\partial \mathbf{u}$, it suffices in the implementation of the above quasi-stationary approach to store only $(\partial \mathbf{c}/\partial \mathbf{u})\mathbf{p}_i^{(k)}$. The decision as to whether Steps 3.*i* and 4.*i* are executed or the stationary approach is used is made via the following statement:

$$\text{if } |\Delta h_i^{(k-1)}| < h_{max} \text{ and } |\Delta h_i^{(k-1)}| < \frac{1}{s}\sum_{i=1}^{s} |\Delta h_i^{(k-1)}| + s_{max}\sigma(|\Delta h_i^{(k-1)}|), \tag{39}$$

then use the stationary approach else execute Steps 3*i* and 4*i*. Here h_{max} and s_{max} are two user-supplied parameters and $\sigma(|\Delta h_i^{(k-1)}|)$ is the standard deviation of the data $\{|\Delta h_i^{(k-1)}| : i - 1, \ldots, s\}$. In this paper the RNR method with strategy (36) is referred to as the quasi-stationary (QS) method. It is evident that the update of the Jacobian by the QS meethod has the *bounded deterioration property*[17] and thus according to the theory the QS method is at least locally linear convergent. Note that $h_{max} \leqslant 0$ results in the RNR method while $s_{max} = h_{max} = \infty$ gives stationary iteration. As we shall see, the most effective strategy is a combination of the QS and Broyden methods where the switch from the QS method to Broyden's method is made after the first few QS steps. It should be noted that the QS method favours implementation on computers with parallel architecture.

NUMERICAL EXAMPLES

All examples presented in this paper were computed using a Pentium PC, 90 MHz. The programme is coded in Microsoft FORTRAN Power Station using double-precision airthmetic with machine epsilon ε_m or order 10^{-16}.

As a test problem we chose the flow over a high-overflow WES-shaped spillway with parameters[9] $L_1 = 7{\cdot}5$, $L_2 = 1{\cdot}615$ and $H_0 = 4{\cdot}13$. The question of whether the point of tangency along the lower nappe is introduced or not is insignificant in the study of the convergence behaviour and thus the point

of tangency is not introduced for the moment. We begin with the study of the interior **h**-iteration level. As already remarked, the **h**-iteration converges only for a certain range of given data Q. It was found that for a coarse mesh discretization the empirical value of the discharge coefficient results in convergent **h**-iteration. In retrospect this greatly facilitates the computation, since any possible failure of the **h**-iteration could be attributed to both poor initial position of Γ_{free} and value of Q. In the case of a fine mesh discretization the empirical value of Q is insufficiently accurate as an initial value of Q. For this reason the discussion of the **h**-convergence is first focused on a computation where a coarse mesh discretization is used. In particular, mesh $M_{2410107}$ (see Table I) was selected as a representative coarse mesh. For a visual comparison between coarse and fine meshes see Figures 2 and 3. Numerical results for three total energy levels $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$ are considered. However, the **h**-convergence behaviour does not notably depend upon H_e and thus, in testing the **h**-convergence only results for $H_e = H_d$ are presented. The corresponding empirical value of the discharge coefficient is $Q_e = 0{\cdot}711$. It turned out that for the chosen mesh discretization Q_e coincides with the critical value, i.e. $\Phi(\Gamma_{\text{free}}(Q_e)) = 0$, and thus the chosen mesh discretization is ideal for testing the variable domain method. The initial position of Γ_{free} is determined as explained in the definition of Ω_r and is shown in Figure 2. Here point D′ of the initial position of Γ_{free} above D is lowered by 0·02. Although the displacement of D′ is minute, it is necessary for convergence of the method. The parameters of the QS method were set to $h_{\max} = 0{\cdot}001$ and $s_{\max} = 2{\cdot}0$. The question of at which iteration step the switch from the QS method to Broyden's method should be made was studied and the results are summarized in Table II. Here the stopping criterion is set to $\varepsilon_{\mathbf{h}} := \max\{\|\Delta\mathbf{h}^{(k)}\|_\infty, \|\mathbf{f}(\mathbf{h}^{(k)})|_\infty\} < 10^{-p}, p = 10$, where $\|.\|_\infty$ is the infinity norm, i.e. $\|\Delta\mathbf{h}^{(k)}\|_\infty = \max_i |\Delta h_i^{(k)}|$. For practical computations it suffices to set $p = 7$. However, in order to be sure that the implementation of the method is free of errors, the overlaid stopping criterion $p = 10$ was used in testing the **h**-iteration code. As can be seen, a switch from the QS method to Broyden's method before step 4 is premature and results in the collapse of the mesh. The longer the switch is postponed, the more robust but also more costly is the method.

Convergence histograms of the RNR method, the QS method and the combined QS and Broyden method with the switch at the fifth iteration step are shown in Figure 4. Note that the RNR and QS methods exhibit quadratic convergence between steps 4 and 8. The reduction of the convergence rate to linear after steps 8 is attributed to the fact that in the finite difference approximation of $\partial\xi_{mN}/\partial h_i$ a fixed step size is used. We report that the computational cost of the RNR method is higher by a factor of 1·5 than that of the combined QS and Broyden method with the switch after the fifth step. An alternative to switching at a fixed iteration step is to switch at the step where $\|\Delta\mathbf{h}^{(k)}\|_\infty$ drops below a certain threshold. The efficacy of the method based upon switching at a fixed iteration step obviously depends

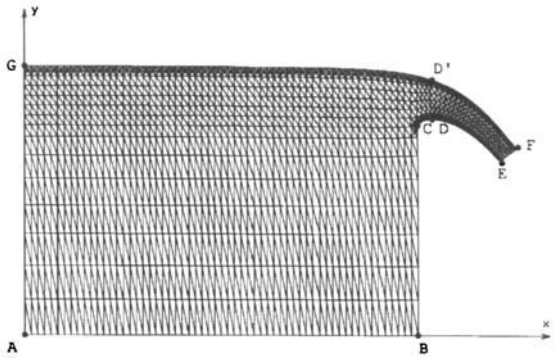

Figure 3. Finite element discretization for mesh $M_{6860320}$. Γ_{free} corresponds to the final position of the critical flow calculation for $H_e = H_d$

Table II. Here k_B is the step at which the switch from the QS method to Broyden's method is made, k_s is the number of iteration steps required for the termination criterion to be satisfied, CPU is the required computational time normalized to the case $k_B = 4$ and n is the number of performed steps (Steps 3.*i* and 4.*i*) of the QS method, as suggested by (39), at the iteration step $k = k_B$. Divergence of the **h**-iteration is denoted by an asterisk

k_B	k_s	CPU	n
1	*	*	37
2	*	*	23
3	*	*	33
4	19	1	21
5	17	1·051	17
6	14	1·062	14
7	11	1·080	12
8	11	1·174	5
9	11	1·253	4
10	11	1·347	3
11	11	1·434	4

upon the quality of the initial position of Γ_{free}. Since during the Q-iteration the quality of the initial position of Γ_{free} varies, switching based on the value of $\|\Delta \mathbf{h}^{(k)}\|_\infty$ was used on all subsequent computations presented in this paper. For a typical threshold value we suggest 10^{-4}. Note that in Figure 4 the chosen threshold value corresponds to switching at the eighth iteration step. In Figures 5(a) and 5(b) we present convergence histograms of the combined QS and Broyden method. Results for four different mesh gradings are given. In all cases the initial position of Γ_{free} is just the final position of Γ_{free} of the coarse mesh $M_{2410107}$ computation, while Q has values of 0·71128331, 0·71031604, 0·71011478 and 0·71011889 for meshes $M_{2430110}$, $M_{4830220}$, $M_{6860320}$ and $M_{10880320}$ respectively. It will be shown later that these values of Q are just critical values of the discharge coefficient for particular mesh discretizations. As can be seen, the convergence rate deteriorates with increasing mesh refinement. This can be attributed to the fact that with increasing mesh refinement the assumption in the computation of the Jacobian that only elements along Γ_{free}, Γ_{in} and Γ_{out} are variable becomes less and less accurate; compare the mesh discretizations in Figures 2 and 3. Notable is the convergence behaviour for mesh $M_{10880320}$. In this case not only is the mesh discretization very fine but also the chosen initial position of Γ_{free} leads to a final position of Γ_{free} which is fairly wavy and

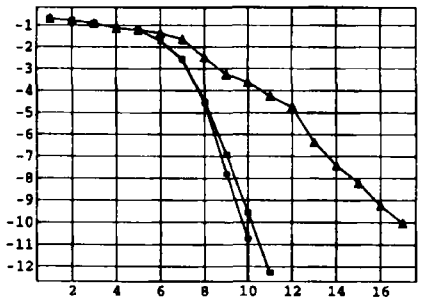

Figure 4. Logarithmic plots of $\|\Delta \mathbf{h}^{(k)}\|_\infty$ versus iteration counter k for RNR method (●), QS method (■) and combined QS and Broyden method (▲)

these together result in some oscillations in the convergence. In particular, $\Phi(\Gamma_{free})$, which measures how wavy Γ_{free} is, has a value of about 73°. Note that in the case of wavy Γ_{free} the approximation that only elements along the variable boundary are movable is even less accurate. It was also found that using a very fine mesh discretization ($n_3 = 80$), different initial positions of Γ_{free} sometimes result in two different final positions. In other words, at a fixed value of Q, multiple solutions to the free boundary value problem exist. Though this is quite awkward, these final positions fortunately have different characters, one being far less wavy than the other, and since a *no-wave* solution is sought, the former is proclaimed as the solution. This of course raises the question of how the initial position of Γ_{free} should be chosen to obtain the solution with *no-wave* character. The answer lies in the refinement process where the initial position of Γ_{free} is updated by replacing it by the final position of Γ_{free}. In particular, if the **h**-iteration with mesh $M_{10880320}$ is initiated with the final position obtained by the critical flow calculation for mesh $M_{7880320}$, a solution with a minute violation $\Phi(\Gamma_{free}) = 0{\cdot}466°$ of the *no-wave* condition is achieved. In the latter case the **h**-convergence is much faster, since it takes only four iteration steps to achieve $\|\Delta\mathbf{h}^{(k)}\|_\infty < 10^{-10}$; see Figure 6, where logarithmic plots of $\|\Delta\mathbf{h}^{(k)}\|_\infty$ versus iteration counter k are given for four different initial values of Q. As can be seen, in all cases the convergence is very satisfactory. We note that owing to the fine mesh discretization the convergence radius of the **h**-iteration with respect to Q is rather narrow, less than 10^{-5}. However, since the critical value of Q obtained with mesh $M_{7880320}$ is a good initial value to start off, the narrowness of the convergence radius of Q will not be a problem. It is concluded from the above discussion that near a critical value of Q the **h**-convergence is stable enough to decide that in the case of an unknown value of

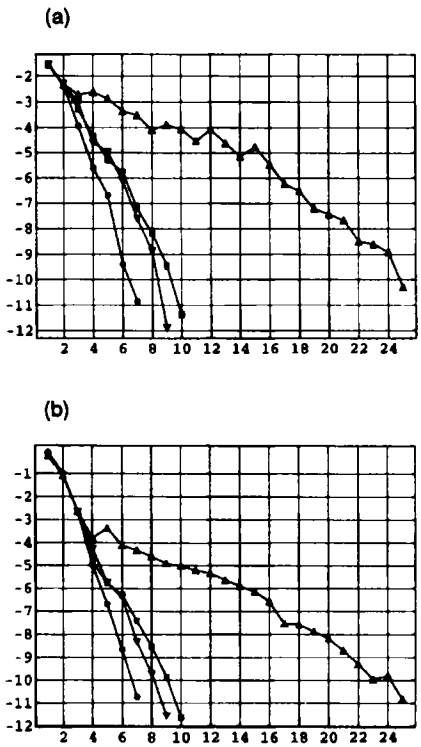

Figure 5. Logarithmic plots of (a) $\|\Delta\mathbf{h}^{(k)}\|_\infty$ and (b) $\|\mathbf{f}(\mathbf{h}^{(k)})\|_\infty$ versus iteration counter k for combined QS and Broyden method. The switch from the QS method to Broyden's method is made after $\|\Delta\mathbf{h}^{(k)}\|_\infty$ falls below 10^{-4}. The meshes are $M_{2430110}$ (●), $M_{4830220}$ (▼), $M_{6860320}$ (■) and $M_{10880320}$ (▲)

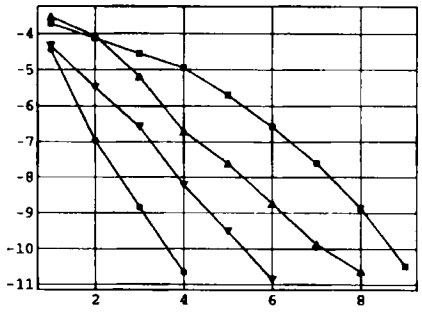

Figure 6. Dependence of **h**-convergence rate upon initial value of Q. The mesh is $M_{10880320}$ and the initial position of Γ_{free} coincides with the final position obtained by the critical flow computation for mesh $M_{7880320}$. $Q = 0{\cdot}71011889$ (●), $0{\cdot}710118$ (▼), $0{\cdot}710115$ (■) and $0{\cdot}710123$ (▲)

Q_c and a relatively good guess of the initial position of Γ_{free} the **h**-convergence is worth pursuing only if $\|\Delta \mathbf{h}^{(k)}\|_\infty$, apart from a few allowed exceptional jumps, decreases monotonically. In the case of jumps of $\|\Delta \mathbf{h}^{(k)}\|_\infty$ the decision as to whether the iteration should be continued or not is facilitated by monitoring the behaviour of $\|\mathbf{f}(\mathbf{h}^{(k)})\|_\infty$; see Figures 5(a) and 5(b).

We now turn to the numerical study of the Q-iteration where the minimum of a function $\Phi(\Gamma_{free}(Q))$ is sought. As soon as an initial value of Q for which the **h**-problem has a solution has been found, the computation of $\Phi(\Gamma_{free}(Q))$ is straightforward. For a new value of Q the associated **h**-problem is initialized by the final position of Γ_{free} of the old value of Q. It turned out in our numerical experiments that in this way the defined $\Phi(\Gamma_{free}(Q))$ is a single-valued function whose definition domain contains some interval with the initial value of Q as an interior point. The topological property of the domain is very important since it allows minimization of $\Phi(\Gamma_{free}(Q))$ as the minimization of a function of a continuously variable Q. However, remember that in the above definition of $\Phi(\Gamma_{free}(Q))$ an important role is played by the initial value of the discharge coefficient, $Q^{(0)}$, as well as the initial position of the free boundary, $\Gamma^{(0)}_{free}$. Explicitly denoting the dependence upon them, we write $(\Gamma_{free}(Q); Q^{(0)}, \Gamma^{(0)}_{free})$. Of course, Φ also depends upon the mesh discretization and thus the full notation is $(\Gamma_{free}(Q); Q^{(0)}, \Gamma^{(0)}_{free}, M_{n_1 n_2 n_3 n_4 n_5})$. The corresponding expanded notations of Q_c and the final position of Γ_{free} are $Q_c(Q^{(0)}, \Gamma^{(0)}_{free}, M_{n_1 n_2 n_3 n_4 n_5})$ and $\Gamma_{free}(M_{n_1 n_2 n_3 n_4 n_5})$. However, on occasions where no ambiguity arises, the simplfied notations $\Phi(\Gamma_{free}(Q))$, Q_c or $Q_c(M_{n_1 n_2 n_3 n_4 n_5})$ and $\Gamma_{free}(Q_c)$ are used. The first step in the minimization of $\Phi(\Gamma_{free}(Q))$ is bracketing of the minima. Since the end points of the interval are not known in advance, this step is very important. How large the first step ΔQ in the bracketing is dependent upon the mesh discretization. For a fine mesh a suggested value is 10^{-6} while for a coarse mesh ΔQ could be of order 10^{-3}. Bracketing is done by stepping downhill until the bracketing triple is found. Here the step size is computed either by magnification of the step size by a constant factor or by using parabolic interpolation.[18] In the case where for a proposed value of Q the **h**-iteration fails to converge, i.e. Q is outside the definition domain, the step size is halved until the **h**-iteration converges. Here a failure of the **h**-iteration is manifested by either collapsing of the mesh or oscillating behaviour of $\|\Delta \mathbf{h}^{(k)}\|_\infty$. As soon as the bracketing triple has been found, minimization is proceeds by Brent's method[18] which utilizes inverse parabolic interpolation.

The stopping criterion for the Q-iteration depends upon the stopping criterion for the **h**-iteration, which is set to $\varepsilon_{\mathbf{h}} < 10^{-7}$. Assuming that the upstream part of Γ_{free} is a straight line, a single vertical displacement Δy of a free boundary node in the upstream part of Γ_{free} changes Φ by

$$\Delta\Phi = 4\frac{180^\circ}{\pi}\tan^{-1}\left(\frac{2n_3\Delta|y|}{L_1}\right). \tag{40}$$

Estimating (40), it follows that for $L_1 = 7{\cdot}5$ and $|\Delta y| < \varepsilon_{\mathbf{h}}$,

$$\Delta\Phi < \frac{180^\circ}{\pi} n_3 \varepsilon_{\mathbf{h}}. \tag{41}$$

For $n_3 \leqslant 80$ and $\varepsilon_{\mathbf{h}} < 10^{-7}$, Φ thus has at least three significant decimal places. In the general case the **h**-displacements might be spread all along Γ_{free} and also Γ_{free} might have several crests and troughs. However, since in the final stage of the **h**-iteration the displacements are localized to the wavy part of Γ_{free}, the stopping criteron $\varepsilon_{\mathbf{h}} < 10^{-7}$ again guarantees that for the *no-wave* solution Φ has at least three significant decimal places. Now, assuming that $\Phi(\Gamma_{\text{free}}(Q))$ is a smooth function, it follows that near Q_c,

$$\Phi(\Gamma_{\text{free}}(Q)) \approx \Phi(\Gamma_{\text{free}}(Q_c)) + \frac{1}{2}\left(\frac{d^2\Phi(\Gamma_{\text{free}}(Q))}{dQ^2}\right)_{Q=Q_c} (Q - Q_c)^2. \tag{42}$$

Estimating $d^2\Phi(\Gamma_{\text{free}}(Q))/dQ^2$ at Q_c for particular numerical results (see Figures 7(a)–7(c)), we found that the second derivative of Φ at Q_c is of the order 10^{12}. Hence it follows from (41) and (42) that for $\varepsilon_{\mathbf{h}} < 10^{-7}$ and $n_3 \leqslant 80$ the appropriate stopping criterion on ΔQ is

$$|\Delta| < \left|\frac{2\Delta\Phi}{d^2\Phi(\Gamma_{\text{free}}(Q))/dQ^2}\right|^{1/2} \approx 10^{-8}. \tag{43}$$

On the other hand, the lower bound of $\Phi(\Gamma_{\text{free}}(Q))$ is known and thus the Q-iteration is instantly terminated if $\Phi(\Gamma_{\text{free}}(Q)) = 0$ within machine epsilon precision.

Convergence histograms for the Q-iteration are shown in Figures 7(a)–7(c). Here $\Phi(\Gamma_{\text{free}}(Q))$ against Q is plotted as a smooth interpolating function which joins values of $\Phi(\Gamma_{\text{free}}(Q))$ computed at steps of the Q-iteration. Numbers below the graph indicate at which iteration step a particular value of $\Phi(\Gamma_{\text{free}}(Q))$ obtained. In the final stage of the Q-iteration the values of $\Phi(\Gamma_{\text{free}}(Q))$ are so clustered together that only the last iteration step is explicitly numbered. Each figure also shows the initial value of Q, while the initial positions of Γ_{free} are $\Gamma_{\text{free}}(M_{2410107})$ for Figures 7(a) and 7(b) and $\Gamma_{\text{free}}(M_{7880320})$ for Figure 7(c). Note in Figure 7 how the bracketing is done; for example, in Figure 7(c) the bracketing triple consists of values of Q at steps 4, 5 and 6. Observe that in this particular case the step size of the fourth Q-iteration step is reduced, since the originally proposed Q falls outside the definition domain of $\Phi(\Gamma_{\text{free}}(Q))$.

As already noted, the Q-iteration is terminated immediately if $\Phi(\Gamma_{\text{free}}(Q)) = 0$. In this case the termination criterion on δQ is not satisfied and thus the question of whether such a value of Q is unique arises. Investigating the neighbourhood of Q for which $\Phi(\Gamma_{\text{free}}(Q)) = 0$, we found that for certain coarse mesh discretizations the length of the interval included in the inverse image of $\Phi(\Gamma_{\text{free}}(Q)) = 0$ is several orders greater than the stopping criterion on ΔQ. In particular, we studied meshes $M_{2405105}$, $M_{2410107}$ and $M_{2420110}$ and energy levels $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$; see Table III. As can be seen, the length of the maximal interval included in $\Phi^{-1}(0)$ contracts notably in the refinement from $M_{2410107}$ to $M_{2420110}$. With further refinements the contraction of the interval continues such that for values of the mesh parameter $n_3 \geqslant 30$ the uniqueness of the solution of $\Phi(\Gamma_{\text{free}}(Q)) = 0$ is guaranteed within the precision of Φ. A solution to $\Phi(\Gamma_{\text{free}}(Q)) = 0$ does not exist for all possible mesh discretizations. In these cases Brent's algorithm in general gives only the local minimum, which is according to our numerical experience unique and thus the global one.

For the design analysis of spillway flows the knowledge of Q_c within a precision of three decimal places is more than sufficient. However, from the viewpoint of the finite element discretization the results are proven to be correct only if their consistency can be shown by mesh refinement. The minimization of $\Phi(\Gamma_{\text{free}}(Q))$ was subjected to such study and the results are summarized in Table IV.

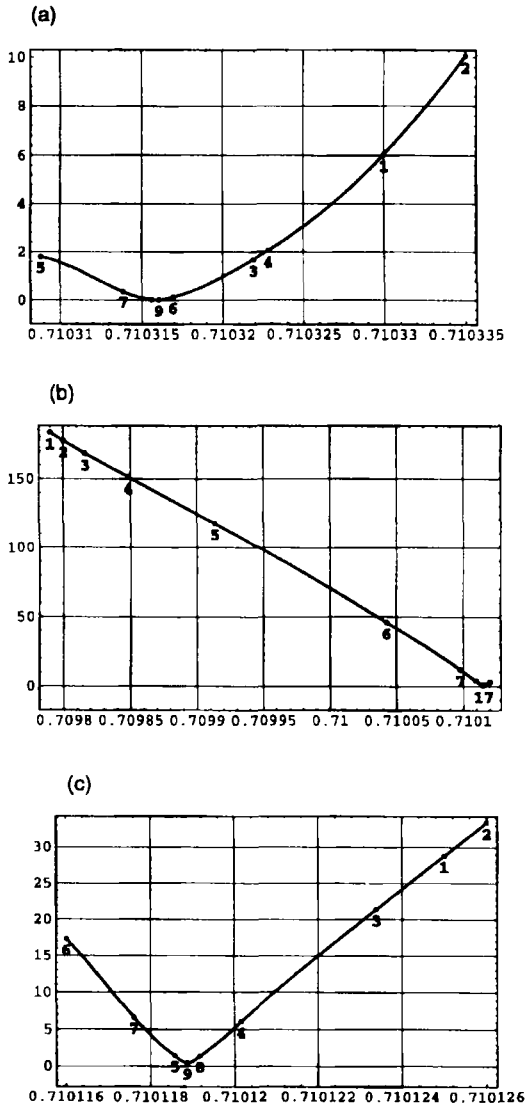

Figure 7. Convergence histograms of Q-iteration level for meshes (a) $M_{4830220}$, (b) $M_{6860320}$ (c) $M_{10880320}$. Total CPU times are 5, 29 and 21 min for meshes $M_{4830220}$, $M_{6860320}$ and $M_{10880320}$ respectively

Table III. Study of the inverse image of $\Phi(\Gamma_{free}(Q)) = 0$ for meshes $M_{2405105}$, $M_{2410107}$ and $M_{2420110}$ by means of finding a maximal interval $[Q_{min}, Q_{max}]$ included in $\Phi^{-1}(0)$. Q_e is the empirical value of the discharge coefficient. For $H_e = 1{\cdot}33$ and meshes $M_{2410107}$ and $M_{2420110}$ the inverse image is the empty set

	$M_{2405105}$		$M_{2410107}$		$M_{2420110}$		
H_e	Q_{min}	Q_{max}	Q_{min}	Q_{max}	Q_{min}	Q_{max}	Q_e
$0{\cdot}5H_d$	0·2277901	0·2306272	0·2266132	0·2303810	0·2284540	0·2284867	0·226
H_d	0·7069500	0·7152800	0·7098814	0·7120455	0·7113255	0·7114037	0·711
$1{\cdot}33H_d$	1·1341547	1·1462406					1·121

According to our numerical experiments, the *no-wave* condition is usually violated only along the upstream part of Γ_{free} and thus the question of whether the observed upstream waves have any meaningful interpretation was studied by varying n_3. Of course, with changing n_3 the structure of the mesh varies, so in order to have meshes of about the same structure, other mesh parameters were also varied. For $H_e = 0{\cdot}5H_d$ the finest mesh used is $M_{4840220}$, since for a greater value of n_1 the elements along the spillway crest are so thin that the Jacobian of the isoparametric transformation becomes ill-conditioned. In all cases except $H_e = H_d$, $n_3 = 80$ and $n_1 \geqslant 8$ the final positions of mesh $M_{2410107}$ serve as satisfactory initial guesses of Γ_{free}, i.e. $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{2410107})$. Moreover, the aforementioned dependence of $\Gamma_{free}(Q_c)$ upon $\Gamma_{free}^{(0)}$ was found only for $H_e = H_d$, $n_3 = 80$ and $n_1 \geqslant 6$. The Q-iteration for meshes $M_{n_1 880320}$, $n_1 = 6$, 7 and 8, started with $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{2410107})$ and $Q^{(0)} = 0{\cdot}71011$ converges to wavy solutions. In particular, $\Phi(\Gamma_{free}(Q_c))$ has values of 157°, 33° and 109° at $Q_c = 0{\cdot}71005130$, 0·71009907 and 0·71007967 for $n_1 = 6$, 7 and 8. The variation in Q_c is here of such magnitude that for meshes $M_{7880320}$ and $M_{8880320}$ even initial values $Q^{(0)} = Q_c(M_{6880320})$ and $Q_c(M_{7880320})$ and initial positions $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{6880320})$ and $\Gamma_{free}(M_{7880320})$ do not result in convergent Q-iteration. However, taking $Q^{(0)} = 0{\cdot}71013$ and $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{2410107})$, *no-wave* solutions for $M_{6880320}$ and $M_{7880320}$ with $Q_c = 0{\cdot}71012229$ and 0·71011814 result. It is interesting to find that at $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{2410107})$ the Q-iteration converges to wavy solutions even if $Q^{(0)}$ has the above *no-wave* values of Q_c. For $n_1 = 8$, 9 and 10, searching for $Q^{(0)}$ for which $\Gamma_{free}^{(0)} = \Gamma_{free}(M_{2410107})$ gives *no-wave* solutions was too expensive and thus the final position of the *no-wave* solution of $M_{7880320}$ was used for $\Gamma_{free}^{(0)}$ together with the best available value of $Q^{(0)}$. In this way *no-wave* solutions were found as reported in Table IV. Note here how $\Phi(\Gamma_{free}(Q))$ diminishes with increasing n_1. As can be seen in Table IV, Q_c is most affected by a change in n_4. This is not surprising, since an increment of n_4 introduces a greater number of new elements per unit length in the x-direction. n_2 affects the mesh within a region which is not of primal importance to the free surface computation and thus n_2 is restricted to values of $n_2 = 4$, 8 and 12. On the other hand, n_5 influences the mesh discretization of a region which is very important. However, it was found that Q_c is not particularly sensitive to n_5 and hence only values of $n_5 = 10$ and 20 are considered. Summing up the results in Table IV, we see that with increasing mesh

Table IV. Values of Q_c for different mesh discretization

	$H_e = 0{\cdot}5H_d$			$H_e = H_d$			$H_e = 1{\cdot}33H_d$		
Mesh	Q_c	Φ_0	Φ	Q_c	Φ_0	Φ	Q_c	Φ_0	Φ
M_{243110}	0·22835670	51·480°	0·000°	0·71128331	57·810°	3·960°	1·14077214	62·300°	7·420°
$M_{2430210}$	0·22782232	51·470°	0·000°	0·71044047	57·816°	3·989°	1·13974275	62·326°	7·458°
$M_{2830210}$	0·22782000	51·468°	0·000°	0·71035172	57·613°	3·805°	1·13952210	62·175°	7·309°
$M_{4830210}$	0·22783583	51·449°	0·000°	0·71031824	53·855°	3·805°	1·13949093	55·659°	0·694°
$M_{4830220}$	0·22783572	51·667°	0·000°	0·71031604	54·002°	0·000°	1·13946149	55·808°	0·700°
$M_{4840220}$	0·22783000	51·666°	0·000°	0·71033123	54·889°	0·877°	1·13947437	56·109°	1·842°
$M_{6840220}$				0·71035623	54·153°	0·130°	1·13948197	55·303°	0·104°
$M_{8840220}$				0·71036205	54·059°	0·021°	1·13948932	55·165°	0·000°
$M_{8840320}$				0·71011434	54·038°	0·016°	1·13919181	55·153°	0·000°
$M_{6860320}$				0·71011478	54·022°	1·000°	1·13919601	55·146°	0·000°
$M_{6880320}$				0·71012229	57·218°	3·196°	1·13920026	55·152°	0·007°
$M_{7880320}$				0·71011814	55·403°	1·380°	1·13919876	55·152°	0·000°
$M_{8880320}$				0·71011740	55·067°	1·045°	1·13919781	55·152°	0·000°
$M_{9880320}$				0·71011814	54·731°	0·697°	1·13919721	55·154°	0·000°
$M_{10880320}$				0·71011889	54·487°	0·466°	1·13919701	55·155°	0·000°
$M_{10880420}$				0·71001945	54·486°	0·466°	1·13907554	55·154°	0·000°
$M_{101280420}$				0·71001013	54·020°	0·467°	1·13905070	55·154°	0·000°

refinement Q_c converges to certain definite values, thus proving the consistency of the results, For a fine mesh discretization, assessing the oscillation of $\Gamma_{free}(Q_c)$ by means of plotting Γ_{free} is illusive, since the oscillation of Γ_{free} is within the thickness of the plotting line. Therefore the oscillation of $\Gamma_{free}(Q_c)$ is represented by graphs of discrete curvature values at free boundary nodal points; see Figures 8(a) and 8(b). Here the abscissa coincides with the x-axis of the computational domain while the c●linate gives the discrete curvature in degrees. The figures show very clearly that the wavelength of the oscillation is proportional to the node spacing. Note that oscillation of the discrete curvature implies oscillation of Γ_{free} only if the sign of the discrete curvature alternates. With refinement form $M_{6880320}$ to $M_{8880320}$ the most oscillating part of Γ_{free} shifts from $x=4$ to 2. This shows very convincingly that the oscillation of Γ_{free} is only due to the discretization and does not have any intrinsic meaning. In discussing Q-iteration, we mainly restricted ourselves to $H_e = H_d$. The other two cases are computationally simpler: no multiple solutions were found, $H_e = 0{\cdot}5H_d$ always results in zero violation of the negative curvature condition, while $H_e = 1{\cdot}33H_d$ is a showcase of how mesh refinement damps out the oscillation of Γ_{free}. Comparison of Q_e and Q_c is very promising. For $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$ the error is less than 1%, 0·1% and 1·7% respectively.

So far we have been concerned only with numerical aspects of the work. On the other hand, the computation of critical flow is a problem of practical interest and the remainder of the paper is devoted to this aspect. In the design of a spillway crest the avoidance of negative pressures on the crest is one of the most important objectives and thus theoretical prediction of the pressure comes as a great help. Potential theory has severe limitations: it cannot model separation and turbulence, which are both very likely to develop behind curvature discontinuities of the crest. However, after the highest point of the crest no separation and further development of turbulence are expected and here the potential model should have predictive power. This is confirmed in Figure 9, which gives a comparison between

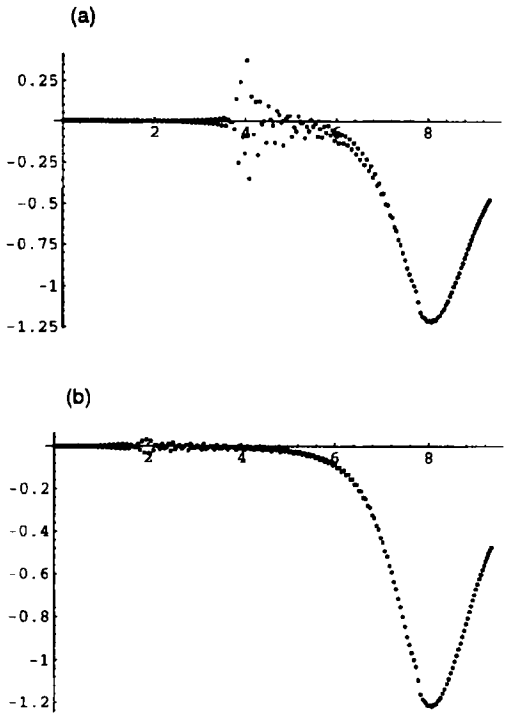

Figure 8. Discrete curvature of Γ_{free} in degrees: (a) Γ_{free} ($M_{6880320}$) and (b) Γ_{free} ($M_{8880320}$)

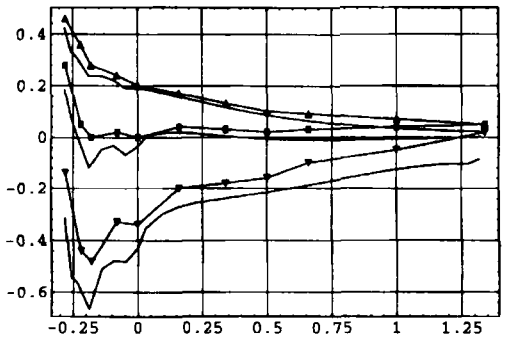

Figure 9. Comparison of pressure along crest. Curves with ▲, ■ and ▼ represent experimental results for $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$ respectively. Corresponding numerical results are given by simple curves which lie below experimental results

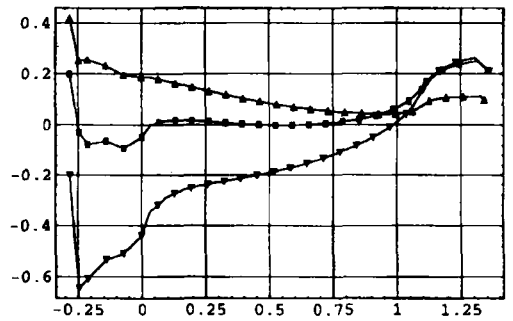

Figure 10. Computed pressure along crest for WES high-overflow spillways with point of tangency included: $H_e = 0{\cdot}5H_d$ (▲), H_d (■) and $1{\cdot}33H_d$ (▼). Marked points now denote computed free boundary nodes. Since after $\xi = 0$ the nodes come too close to each other, only half the nodes after $\xi = 0$ are marked

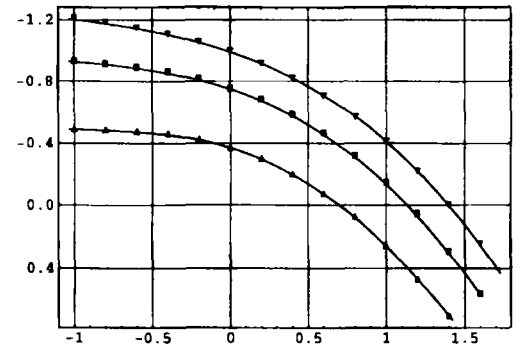

Figure 11. Comparison of experimental and computed positions of Γ_{free}. Experimental data for $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$ are marked with ▲, ■ and ▼ respectively. Computed positions are given by simple curves

theoretical and experimental[11] results for the pressure distribution along the crest. Here the abscissa is the local crest co-ordinate ξ, where the highest point of the crest comes at $\xi = 0$. The comparison is to some degree hampered by the fact that in the available experimental data approaching velocity head is neglected. If it were included, experimental data would be shifted downwards by a small amount. Even after this correction, however, discrepancies, especially in the case of $H_e = 1{\cdot}33H_d$, still remain. The more pronounced discrepancies at larger heads are attributed to the fact that separation and turbulence then becomes more important. Note that the higher values of the experimental results are in agreement

with the theory of boundary layer separation.[19] Nevertheless, it is remarkable how the numerical results follow the experimental data. The present numerical results come close to the results obtained by the Dressler approximation,[10] which accounts for bed curvature in the shallow flow equations. However, in the particular example, the Dressler approximation is of a more restrictive nature since it is valid only for $\xi \geqslant 0{\cdot}1$. The influence of the shape of the spillway crest upon the pressure distribution is demonstrated in Figure 10, where the point of tangency at $\xi = 1{\cdot}096$ is introduced. How this affects the flow depends upon H_e. For $H_e = 0{\cdot}5H_d$ the pressure is notably influenced upon upstream of $\xi = 1{\cdot}0$, while for $H_e = 1{\cdot}33H_d$ it is affected up to $\xi = 0{\cdot}5$. This is attributed to the greater change in the position of Γ_{free} due to the greater bulk of the fluid for higher values of H_e. According to this argument, we conclude that correct positioning of Γ_{out} is more important for larger values of H_e. A comparison between computed and measured positions of Γ_{free} is given in Figure 11. Meshes $M_{4840220}$, $M_{101280420}$ and $M_{101280420}$ were used for $H_e = 0{\cdot}5H_d$, H_d and $1{\cdot}33H_d$. As can be seen, the comparison is quite satisfactory, especially for $H_e = 0{\cdot}5H_d$. We note that our computed positions come much closer than the positions from Reference 9.

CONCLUSIONS

A numerical computation of the critical flow over a spillway by minimization of the oscillation of the free boundary has been presented. The method has two iteration levels. It has been shown that with increasing mesh refinement the Q-convergence radius of the inner iteration rapidly shrinks. This suggests that in the limit of mesh discretization the solution to the inner iteration exists only for a unique value of Q which comes as a critical value of the discharge coefficient. It is concluded that the present method is capable of tracing the shrinkage of the Q-convergence radius and hence that it converges to the critical value of the discharge coefficient.

REFERENCES

1. M. J. Diersch, A Schirmer and K. F. Busch, 'Analysis of flows with initially unknown discharge', *J. Hydraul. Div. ASCE*, **103**, 213–232 (1977).
2. A. H. D. Cheng, J. A. Ligget and P. L. F. Liu, 'Boundary calculation of surface and spillway flows', *J. Hydraul. Div. ASCE*, **107**, 1163–1178 (1981).
3. M. J. O'Carrol and E. F. Toro, 'Numerical computations of critical flow over a weir', *Int. j. numer. methods fluids*, **4**, 499–509 (1984).
4. G. Mejak, 'Numerical solution of Bernoulli-type free boundary value problems by variable domain method', *Int. j. numer. methods end.*, **37**, 4219–4245 (1994).
5. G. Mejak, 'Numerical computation of critical flow over a weir', in R. Narishma (chairman), *Collect. Short Abstr. 14th Int. Conf. on Numerical Methods in Fluid Dynamics*, Bangalore, 1994, pp. 140–141.
6. M. Ikegawa and K. Washizu, 'Finite element method applied to analysis of flow over spillway crest', *Int. j. numer. methods eng.*, **6**, 179–189 (1973).
7. P. Bettess and J. A. Bettes, 'Analysis of free surface flows using isoparametric finite elements', *Int. j. numer. methods eng.*, **19**, 1675–1689 (1983).
8. J. M. Aitchison, 'A finite element solution for critical flow over a weir', in D. H. Norrie (ed.), *Proc. 3rd Int. Conf. on Finite Elements in Flow Problems*, Vol. 1, Calgary, 1980, pp. 52–59.
9. W. D. L. Finn and E. Varoglu, 'Variable domain finite element analysis of free surface gravity flows', *Comput. Fluids*, **6**, 103–114 (1978).
10. N. S. Sivakumaran, R. J. Hosking and T. Tingsanchali, 'Steady shallow flow over a spillway', *J. Fluid Mech.*, **111**, 411–420 (1981).
11. V. T. Chow, *Open-Channel Hydraulics*, McGraw-Hill, Tokyo, 1959.
12. P. G. Ciarlet and P. A. Raviart, 'Interpolation theory over curved elements, with application to finite element methods', *Comput. Methods Appl. Mech. Eng.*, **1**, 217–249 (1972).
13. G. Mejak, 'Finite element analysis of axisymmetric free jet impingement', *Int. j. numer. methods fluids*, **13**, 491–506 (1991).
14. R. A. Brockman, 'Geometric sensitivity analysis with isoparametric finite elements', *Commun. appl. numer. methods*, **3**, 495–499 (1987).

15. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
16. J. E. Dennis and J. J. Moré, 'Quasi-Newton methods, motivation and theory', *SIAM Rev.*, **19**, 46–89, 91977).
17. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
18. W. H. Press, S. A. Teukolsky, W. T. Vettering and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd edn, Cambridge University Press, Cambridge, 1992.
19. G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge, 1967.