INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:1231–1255 (DOI: 10.1002/fld.684)

Development and testing of a simple 2D finite volume model of sub-critical shallow water flow

Matthew Horritt*,†

School of Geographical Sciences, University of Bristol, University Road, BS8 1SS Bristol, U.K.

SUMMARY

Many environmental applications of shallow water flow modelling can be characterized as only slowly varying and everywhere sub-critical. A simplified finite volume model is therefore developed that is capable of describing pertinent shallow water flow processes more efficiently than the usual Godunov/ Riemann characteristics approaches. The model is tested against a number of analytical and numerical solutions to the governing equations. The model reproduces accurately flow round a circular bend, flow over topography, flow up an initially dry beach and floodwave propagation down a meandering river reach, with mass conservative solutions. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: finite volume; shallow water; sub-critical flow

1. INTRODUCTION

Finite volume methods have long been popular for the modelling of environmental flows whose behaviour is well described by the two-dimensional (2D) shallow water equations [1–7]. The application of finite volume techniques in this context follows directly from the earlier application of 2D compressible flow for aerodynamic problems, which is made possible because of the close analogy between the equations for 2D compressible flow and shallow water flow. Furthermore, there are phenomena such as wave and shock propagation that can be observed in both aerodynamic and shallow water flows. The treatment of these phenomena is not straightforward and sophisticated numerical schemes are required to avoid inaccurate solutions in their presence, which results in lengthy development and computation times for such methods. It is not clear, however, that such an approach is required for many environmental flows, and as such traditional finite volume approaches may represent a misuse of resources. The specification for the design of a numerical model of shallow water flow is therefore re-appraised in this paper, and the simplest solution to the modelling problem is sought.

*Correspondence to: M. Horritt, School of Geographical Sciences, University of Bristol, University Road, BS8 1SS Bristol, U.K.
†E-mail: Matt.Horritt@bristol.ac.uk

Contract/grant sponsor: NERC

Received 27 December 2002
Revised 22 July 2003

The development of numerical models of environmental flows involves two stages: identification of an equation base suitable for adequately describing significant hydraulic processes, and then development of a numerical scheme for the solution of these equations. Most modelling schemes are based on the 2D shallow water equations, partly for the pragmatic reason that 3D schemes, which may be required to represent secondary flows in compound channels, are computationally intensive, and thus limited to small-scale studies of in bank flows. That convention is followed in this paper, and thus the governing equations for continuity and conservation of momentum are:

$$\frac{\partial h}{\partial t} + \nabla\cdot(\mathbf{u}h) = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{v_t}{h}\nabla(h\nabla\cdot\mathbf{u}) + g\nabla(h+z) + \frac{gn^2|\mathbf{u}|\mathbf{u}}{h^{4/3}} = 0 \tag{2}$$

h is the flow depth, $\mathbf{u} = [u, v]^{\mathrm{T}}$ is a 2D depth-averaged flow velocity vector, z the bed elevation, g the acceleration due to gravity, n Manning's coefficient of roughness and v_t a turbulent eddy viscosity which parameterizes horizontal turbulent momentum transfer. Equation (1) is satisfied by the flow irrespective of its 2- or 3D nature, as long as water is not lost or gained in significant amounts at the fluid's upper and lower boundaries through evaporation, rainfall or interaction with a groundwater system. These processes will contribute source terms with very small magnitudes (typically mm per hour), and therefore any flow model is required to generate mass conservative solutions satisfying (1). The terms of the momentum equation (from left to right) are: acceleration, advection, turbulent momentum transfer, free surface gradient and bed friction. For environmental flows we will often be confronted with rough and vegetated beds, and topographic features spread across many spatial scales. The treatment of topographic and friction source terms is therefore of vital importance in this application. These are terms which are often neglected in finite volume models where the emphasis lies in the treatment of shock phenomena. Flows with steep velocity gradients (which can be topographically induced) will also generate significant advection terms (for example in meandering channel flow [8]), and so this term must be included. Time derivative terms will be important in reproducing dynamic behaviour such as floodwave propagation and tidal flows in response to changing boundary conditions. Low topographic slopes on floodplains and the presence of tidal flats may produce a moving or free boundary problem, and so some method of dealing with a moving or uncertain shoreline is also required. The turbulence term (here represented by a simple eddy viscosity model) may also be important, for example if a shear zone develops between the channel and floodplain [9], but is difficult to quantify [10] especially in the presence of vegetation [11]. Turbulence is also likely to be highly three-dimensional in nature and hence a 2D model is unsuitable for modelling flows at scales where this interaction is important. This term is therefore neglected in this study, but remains an issue for future research when further laboratory and field experiments allow its measurement and parameterization for shallow water flow models. The presence of vegetation may also introduce vertical velocity structures which depend strongly on plant properties and therefore Manning's equation may not be appropriate for modelling these interactions, and again a fully 3D model may be required.

Having identified the processes to be represented in a 2D flow model, we now aim to solve the governing equations using a suitable numerical technique. The necessity of including processes across a range of spatial scales means that techniques capable of operating on

unstructured meshes [12] will be more appropriate than those such as the finite difference method which rely on structured and often regular meshes. The finite element method has been used with irregular meshes of triangular or quadrilateral elements to model floodplain [13–16] and estuarine [17, 18] flows, but although the finite element method can be shown to be globally absolutely mass conservative [19] it may still produce solutions with local mass conservation errors in some implementations [8]. The finite volume method is therefore adopted here. The traditional finite volume approach starts by casting the governing equations into a form relating the time derivative of the solution field to the divergence of associated fluxes, with source terms taken into account:

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \\ hv \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{bmatrix} + S = 0 \tag{3}$$

S is a vector of source terms due to bed slope, bed friction, etc. When this equation is averaged over a finite volume element and the divergence theorem applied, the rate of change of the unknowns h, hu and hv, is related to the normal fluxes across the boundary of the element:

$$A_i \frac{\partial \mathbf{U}_i}{\partial t} = -[(F_n)_{ij}L_{ij} + (F_n)_{ik}L_{ik} + (F_n)_{il}L_{il}] - \int_{A_i} S \, \mathrm{d}A \tag{4}$$

$\mathbf{U}_i = [h, hu, hv]^{\mathrm{T}}$, etc. are the solution vectors defined at the centroids of the elements, F_n are the normal fluxes at each edge, L_{ij}, etc. are the lengths of the edges (Figure 1), and A_i is the area of the element. Since the unknowns are represented as point values at the centroid of the element, the finite volume technique hinges on an appropriate interpolation method to express the fluxes at the cell boundaries. The crux of a finite element model is thus devising an interpolation method that ensures conservation of mass and momentum between time steps. The problem is not trivial: the fluxes must be interpolated at edges where the

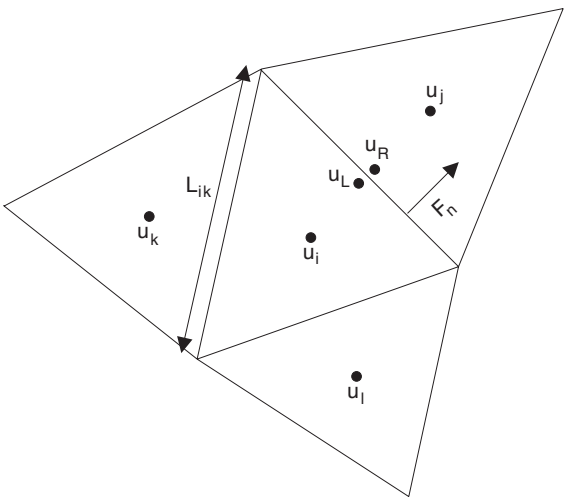

Figure 1. Notation for flux calculations at element edges.

solution is undefined because of the piecewise constant nature of the discretised variables, and the problem is exacerbated by the non-linearity of the fluxes' dependence on the solution.

The interpolation techniques presented so far in the literature are manifold. Most use the Godunov approach, whereby the calculation of the fluxes is transformed into a discontinuous initial value problem in a 1D local co-ordinate system normal to the cell edge, called the Riemann problem. There are many approaches to solving the Riemann problem, the solver of Roe [20] being one of the most popular [2, 3, 5, 6, 21–23]. According to the Roe scheme, the normal flux F_n at the boundary can be written in terms of the fluxes on the left and right sides of the edge (F_L and F_R) and the solution values at the cell centres to the left and right of the edge (U_L and U_R):

$$F_n = \tfrac{1}{2}(F_L + F_R) + |A|(U_L - U_R) \tag{5}$$

The flux Jacobian matrix $A = \partial F/\partial U$ is then evaluated using appropriate mean values of the solutions to the left and right of the edge. The Roe solver has been used to good effect for shallow water flow solvers, generating mass conservative solutions that predict well the propagation of hydraulic jumps. Other techniques have been developed to solve the Riemann problem in a finite volume/shallow water context, such as the scheme of Osher [24, 25] and the Harten–Lax–van Leer scheme [26, 27]. Other numerical ruses such as flux limiters [2, 21] and various upwinding schemes [28, 29] also need to be used to avoid spurious numerical oscillations and under- or overshoot near shocks.

Such techniques represent effective generalized methods for generating solutions of the shallow water equations, with a solid grounding in characteristics theory, and are capable of reproducing hydraulic jumps without mass balance errors or spurious oscillations in the solution, as long as appropriate numerical techniques are used. Indeed, without any *a priori* knowledge of the nature of the solutions, these sort of techniques will be the best to use. However, because of the significant computational burden imposed by these numerical techniques, their application in scenarios where, for example, the sub-critical nature of the flow is observed may be wasteful. The solutions to most lowland fluvial flood problems for example are sub-critical. It is clear therefore that there is a class of environmental flows for which an approach simpler to the shock capturing techniques discussed above can be adopted to good effect, as long as the relevant hydraulic processes discussed above are incorporated. A simple finite volume (SFV) model for this class of problems is therefore presented and tested in this paper. Section 2 describes the development of the model, which is tested against analytical solutions of the shallow water equations, and mass balance errors measured, in Section 3. Results are discussed and summarized in Sections 4 and 5.

2. MODEL DEVELOPMENT

A discretized solution to (1) and (2) is sought over an unstructured mesh of triangular elements, for each of which the area averaged quantities h_i, $(hu)_i$ and $(hv)_i$ are defined. Thus, for a mesh of N elements the model will solve for a vector of $3N$ unknowns at each time step. The treatment of the continuity equation, the momentum equation, boundary conditions, time development and wetting and drying are now described.

2.1. Continuity equation

Integrating the equation over a finite volume Ω_i with boundary Γ_i and outward normal unit vector $\mathbf{n}$, and applying the divergence theorem, gives

$$\frac{\partial h_i}{\partial t} + \frac{1}{A_i}\int_\Gamma (\mathbf{u}h)\cdot\mathbf{n}\,\mathrm{d}\Gamma = 0 \tag{6}$$

For the triangular elements used here the integral around the element is written as the sum of the contributions from each edge (with notation as depicted in Figure 1):

$$\frac{\partial h_i}{\partial t} + \frac{1}{A_i}((F_n)_{ij}L_{ij} + (F_n)_{ik}L_{ik} + (F_n)_{il}L_{il}) = 0 \tag{7}$$

The normal fluxes are the scalar product of $\mathbf{u}h$ and the outward normal to the edge, $(F_n)_{ij} = (h\mathbf{u})\cdot\mathbf{n}_{ij}$, etc. Two methods are used for the interpolation of $h\mathbf{u}$. First-order interpolation simply uses the mean of the values to the left and right of the edge:

$$(F_n)_{ij} = \frac{1}{2}((hu_i + hu_j)(n_x)_{ij} + (hv_i + hv_j)(n_y)_{ij}) \tag{8}$$

A second-order scheme uses the values at surrounding nodes to make an improved estimate of the value of the variable at the cell edge (Figure 2) [21]. The interpolated depth, for example, is written as

$$h_\mathrm{L} = h_i + \mathbf{r}\cdot\nabla h = h_i + r_x\frac{\partial h}{\partial x} + r_y\frac{\partial h}{\partial y} \tag{9}$$

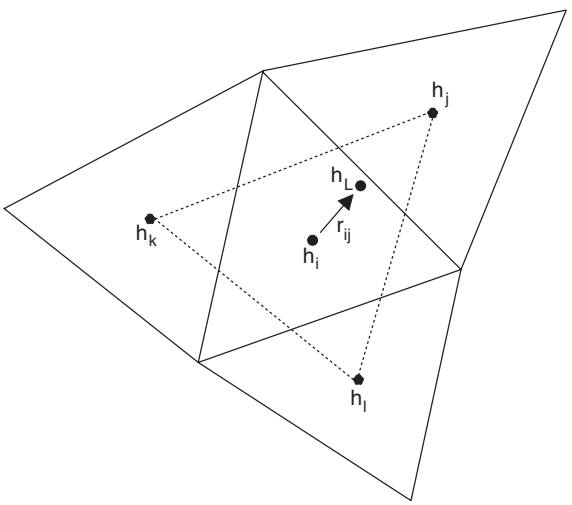

Figure 2. Second-order interpolation of flow depth h. $\mathbf{r}_{ij}$ is the displacement vector between the cell centre and the mid-point of the edge. The local gradient is calculated from the neighbouring elements. A' is the area of the triangle drawn with dashed lines.

where **r** represents the displacement between the cell centre and the cell edge in question. The gradient is estimated from the values at surrounding nodes using a planar approximation, e.g.:

$$\frac{\partial h}{\partial x} \approx \frac{1}{2A'}(h_j(y_k - y_l) + h_k(y_l - y_j) + h_l(y_j - y_k)) \tag{10}$$

A' is the area of the triangle joining neighbouring cell centres, h_j, h_k and h_l are the depths at the three neighbouring cells, and y_j, y_k and y_l are the y co-ordinates of the centres of the neighbouring cells. The first-order scheme uses no information about the form of the finite volume mesh when interpolating variables, and its accuracy decreases for irregular meshes containing elements of various sizes. The second-order scheme gives more accurate estimates of the integrated gradients which are far less dependent on the arrangement of the mesh.

2.2. *Momentum equation*

In integrating the momentum equation over a finite volume, the advection term is the most troublesome. The bed elevation and depth gradients can be treated by the same first- and second-order schemes as the flow divergence in the continuity equation. Using the same scheme for the bed elevation and depth gradients also ensures that a horizontal water free surface generates no spurious velocities that can arise from differences in the treatment of these two terms [3, 30]. The friction term is easily dealt with by replacing **u** and h with their element-averaged quantities. The advection term requires special attention.

The term is rewritten in component form

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = \begin{bmatrix} u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} \\ u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} \end{bmatrix} \tag{11}$$

The piecewise constant nature of u and v is then used to make an approximation to the integral of the advection terms over an element, for example the x-component becoming

$$\int_\Omega \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) \mathrm{d}\Omega \approx u_i \int_\Omega \frac{\partial u}{\partial x}\,\mathrm{d}\Omega + v_i \int_\Omega \frac{\partial u}{\partial y}\,\mathrm{d}\Omega = u_i \int_\Gamma u n_x \,\mathrm{d}\Gamma + v_i \int_\Gamma u n_y \,\mathrm{d}\Gamma \tag{12}$$

This is a non-conservative form of the advection term, in contrast to the conservative forms generally used in finite volume applications. In this respect, the model approach is similar to a first-order finite element approach using piecewise uniform approximation. Averaging the fluxes for the left and right sides of an edge (as for the first-order scheme above) gives the finite volume approximation for the x-component of the advection term as

$$\begin{aligned} &\frac{u_i}{2}(L_{ij}(n_x)_{ij}(u_i + u_j) + L_{ik}(n_x)_{ik}(u_i + u_k) + L_{il}(n_x)_{il}(u_i + u_l)) \\ &\quad + \frac{v_i}{2}(L_{ij}(n_y)_{ij}(u_i + u_j) + L_{ik}(n_y)_{ik}(u_i + u_k) + L_{il}(n_y)_{il}(u_i + u_l)) \end{aligned} \tag{13}$$

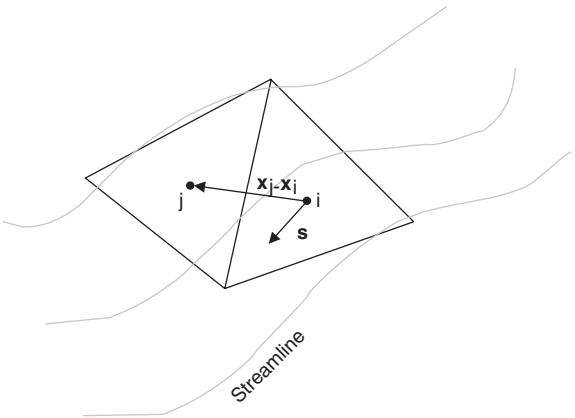

Figure 3. Upwinding for advection terms. In this case **r** is a vector pointing in the upstream direction.

This gives a first-order accurate estimate for the advection term. To promote model stability and prevent spurious oscillations in the solution this term is now upwinded by instead using a weighted mean of the left and right fluxes, for example,

$$\frac{L_{ij}u_i n_x}{2}(\alpha^- u_i + \alpha^+ u_j) + \frac{L_{ij}v_i n_y}{2}(\alpha^- u_i + \alpha^+ u_j) \tag{14}$$

The weighting factors $\alpha^{\pm}$ are defined in terms of the upwind direction **s** and the vector joining the centroids of the two elements, $\mathbf{x}_i$ and $\mathbf{x}_j$ (Figure 3):

$$\alpha^{\pm} = 1 \pm \frac{\mathbf{s}\cdot(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{15}$$

Thus, the flux across the boundary is biased towards j if the jth element lies upwind of the ith element. The upwind direction **s** is taken from the velocity vector at the centre of the cell. The effect of the upwinding is adjusted using the magnitude of **s** ($\in [0,1]$). A magnitude of 0.5 was found to give satisfactory results for a wide range of sub-critical flow conditions.

2.3. *Boundary conditions*

Three types of boundary are defined: a closed (slip wall) boundary, one with an imposed flow rate and one with imposed flow depth. As the flux across an element boundary is calculated in terms of the solutions on either side of the boundary, boundary conditions are used to determine solution values outside the domain. Defining a local co-ordinate system such that u' and v' are velocity components normal to and parallel with the boundary, and using subscripts L and R to denote the solution outside and inside the boundary respectively, the slip wall boundary condition can be written

$$\begin{aligned} h_{\mathrm{L}} &= h_{\mathrm{R}} \\ u'_{\mathrm{L}} &= 0 \\ v'_{\mathrm{L}} &= v'_{\mathrm{R}} \end{aligned} \tag{16}$$

Thus, the boundary condition implies that the depths on either side of the boundary are equal, the component of velocity normal to the edge is zero, and the components parallel to the edge are equal. u' and v' are then transformed back into the global co-ordinate system, allowing expressions such as Equation (8) to be used to calculate the flux across an element edge when there is no neighbouring element on the other side of the edge. The values of h, u and v are also used in the momentum equation. For an open boundary with an imposed flowrate, the boundary conditions are

$$\begin{aligned} h_{\mathrm{L}} &= h_{\mathrm{R}} \\ u'_{\mathrm{L}} &= \frac{q_{\mathrm{imposed}}}{h_{\mathrm{R}}} \\ v'_{\mathrm{L}} &= 0 \end{aligned} \tag{17}$$

For an imposed depth boundary the conditions are

$$\begin{aligned} h_{\mathrm{L}} &= h_{\mathrm{imposed}} \\ u'_{\mathrm{L}} &= u'_{\mathrm{R}} \\ v'_{\mathrm{L}} &= v'_{\mathrm{R}} \end{aligned} \tag{18}$$

These three types of boundary condition are adequate for dealing with a wide range of modelling problems, such as imposed tidal elevations for estuarine modelling, impermeable boundaries for the edge of the domain and imposed fluxes, for example from depth and velocity measurements for fluvial modelling. Imposed total flux (e.g. from a rated section) can be dealt with by partitioning the flux appropriately between boundary elements.

2.4. Time development and solution

Time derivative terms are discretized using finite differences in an implicit scheme, which can then be combined with the spatial integration of the fluxes, e.g. from Equation [7]. The continuity equation for example becomes for element i with neighbours j, k and l:

$$\begin{aligned} &\frac{h_i^{n+1} - h_i^n}{A_i \Delta t} + \frac{\theta}{2}((h_i^n u_i^n + h_j^n u_j^n)(n_x)_{ij} + (h_i^n v_i^n + h_j^n v_j^n)(n_y)_{ij}) \\ &+\frac{\theta}{2}((h_i^n u_i^n + h_k^n u_k^n)(n_x)_{ik} + (h_i^n v_i^n + h_k^n v_k^n)(n_y)_{ik}) \\ &+\frac{\theta}{2}((h_i^n u_i^n + h_l^n u_l^n)(n_x)_{il} + (h_i^n v_i^n + h_l^n v_l^n)(n_y)_{il}) \\ &+\frac{(1-\theta)}{2}((h_i^{n+1} u_i^{n+1} + h_j^{n+1} u_j^{n+1})(n_x)_{ij} + (h_i^{n+1} v_i^{n+1} + h_j^{n+1} v_j^{n+1})(n_y)_{ij}) \\ &+\frac{(1-\theta)}{2}((h_i^{n+1} u_i^{n+1} + h_k^{n+1} u_k^{n+1})(n_x)_{ik} + (h_i^{n+1} v_i^{n+1} + h_k^{n+1} v_k^{n+1})(n_y)_{ik}) \\ &+\frac{(1-\theta)}{2}((h_i^{n+1} u_i^{n+1} + h_l^{n+1} u_l^{n+1})(n_x)_{il} + (h_i^{n+1} v_i^{n+1} + h_l^{n+1} v_l^{n+1})(n_y)_{il}) = 0 \end{aligned} \tag{19}$$

Subscripts refer to element numbers, superscripts to time steps and θ is an implicitation factor which varies between 0 for an explicit model and 1 for a fully implicit model. Similar expressions are found for the time development of the momentum equation. Throughout the rest of the paper a default value of 1 is assumed for θ, giving a fully implicit method. The non-linear system is solved for iteratively using Newton's method, each iteration yielding a linear system with coefficients given by the Jacobian matrix. This linear system is solved using an iterative generalized minimum residual (GMRES) solver [31]. This avoids the need to invert the Jacobian matrix (a costly $O(N^3)$ process), and we need only compute the product of the Jacobian and a vector in the model subspace. This is performed using a second-order accurate finite difference approximation. For example, expanding the function $\mathbf{g}$ about $\mathbf{u}$, the product $\mathbf{J}\cdot\mathbf{w}$, where $\mathbf{J} = \partial\mathbf{g}/\partial\mathbf{u}$ and $\mathbf{w}$ is an arbitrary vector, is approximated as

$$\mathbf{J}\cdot\mathbf{w} \approx \frac{\mathbf{g}(\mathbf{u} + \varepsilon\mathbf{w}) - g(\mathbf{u} - \varepsilon\mathbf{w})}{2\varepsilon} \tag{20}$$

where ε is some number chosen to be small enough to make the finite difference approximation accurate, but large enough to avoid the effects of rounding errors in the calculation. Thus, there is no requirement to store the full Jacobian, making a significant saving in storage and coding is simplified since the elements of the Jacobian do not have to be found explicitly. This second order approximation is preferable to the first-order equivalent, still only requiring two function evaluations, but allowing a much larger value of ε to be used and thus reducing rounding errors considerably. The solution scheme is $O(N^{1.3})$ in time and $O(N)$ in storage, allowing large meshes to be used. The model is stable with time steps up to $\sim$50 s for typical fluvial applications with elements a few 10 m across, although in some cases longer time steps do not necessarily lead to faster computation if the number of Newton–Raphson and GMRES iterations increases significantly.

2.5. *Wetting and drying*

The presence of low topographic slopes in floodplain and estuarine environments mean that many situations involve a moving boundary problem, or in the case of steady-state calculations a free boundary problem. In this case we deal with the wet/dry boundary using a fixed computational mesh, and use an algorithm to deal with elements switching between wet and dry states. Three types of elements can be identified [19] (Figure 4): a tidal wetting element, a drying element and a dam break element. In keeping with the simple approach adopted so far, rather than using a treatment of the Riemann problem with zero depth on one side [2], a simple element masking scheme is used, similar to that of Reference [3]. Dry elements are excluded from the computation, and the wetting and drying algorithm is implemented at each time step to update the element mask, and to redistribute water in vicinity of the shoreline if necessary to ensure mass conservation.

A tidal wetting element is detected when an element has a higher free surface but lower bed elevation than its dry neighbour. The new water depths in the two cells are then calculated to give the same free surface elevation in the two cells and the same total water volume as before:

$$h_i^{n+1} + z_i = h_j^{n+1} + z_j \tag{21}$$

$$h_i^n A_i + h_j^n A_j = h_i^{n+1} A_i + h_j^{n+1} A_j \tag{22}$$

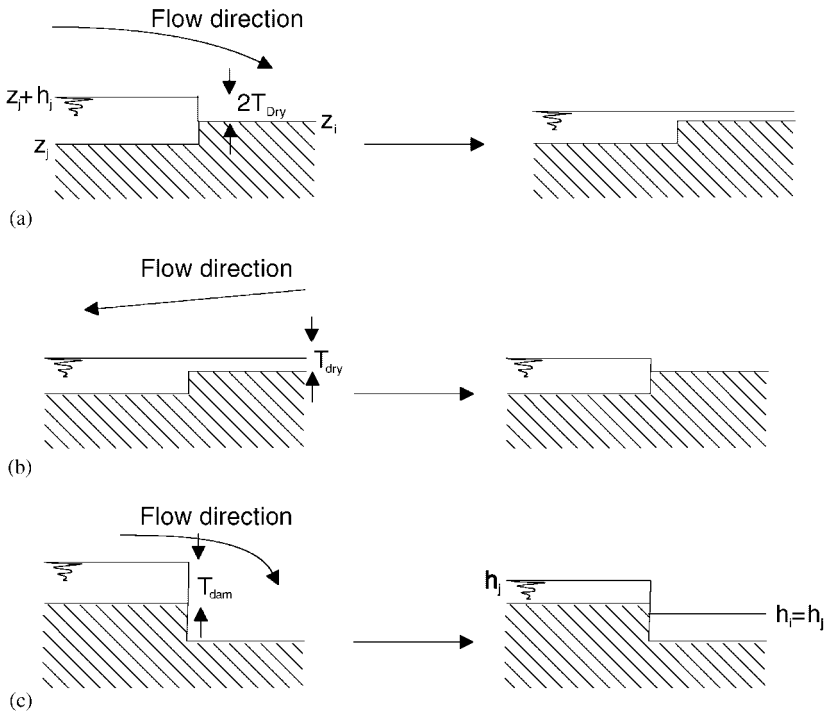

Figure 4. Three types of wetting/drying elements. In a tidal wetting element (a), when the free surface in the wet element exceeds the bed elevation of the dry element by $2T_{\mathrm{Dry}}$, the water in the wet cell is distributed between the two cells to give a horizontal free surface. In a drying cell (b), the water is removed when the depth drops below T_{Dry} and redistributed to neighbouring wet cells. In a dam break element (c), when the depth of a shoreline cell exceeds T_{Dam} the water is redistributed to give equal water depths in the cells.

These simultaneous equations can then be solved for the unknown updated depths h_i^{n+1} and h_j^{n+1}. The flow velocity in the newly wetted cell is set to 0, and the velocity in the previously wet cell is unaffected. Model stability is improved and the oscillation of cells between wet and dry states prevented if it is ensured that the new water depth in the previously dry cell is above the drying threshold (see below), as this prevents a cell oscillating between wet and dry states at every time step. In practice this is done by only wetting the element if the free surface height of the wet element exceeds the bed elevation of the dry elements by approximately twice the drying threshold, so that h_i^{n+1} is greater than the drying threshold. If the dry element has two wet neighbours, then (21) and (22) are solved using the wet element with the highest free surface. As with any model dealing with depths tending to zero at the shoreline, there is always the problem of the flow regime in an element switching from super- to sub-critical as the flow depth increases. This may lead to numerical instabilities, especially for this model which is not designed to handle super-critical flows. In practice this problem was not observed in any of the test cases covered in this paper, but may manifest itself in modelling other flow conditions.

A drying element is simply identified when its water depth drops below a threshold, T_{Dry}. The element is then masked and the water contained moved to the neighbouring wet element with the lowest free surface. The threshold value required depends on the rate at which water levels change and the model time step, as too small a value will mean that a wet element may dry out completely in one time step. The threshold is given therefore by

$$T_{\mathrm{Dry}} > \frac{\partial h}{\partial t}\,\Delta t \tag{23}$$

$\partial h/\partial t$ can be estimated from the model boundary conditions, but since topographic features may induce rapid changes in depth this may need to be adjusted through trial and error. Typical values are 10^{-4}–10^{-3}.

Although the SFV model is not designed for simulating dam breaks, dam break-type wetting elements, where a wetting front propagates down slope, may occur in flood and estuarine flows due to uneven topography. Dam break-type elements may therefore have to be dealt with even for problems which are chiefly made up of tidal wetting elements, and dam break elements make up a small minority of those on the wetting front. These are identified when a wet element with water depth above a threshold T_{Dam} is next to a dry element with a lower bed elevation. In this case, a similar scheme to that used for the tidal wetting element is employed, with the volume of water in the two elements being conserved, but the water depths (rather than free surface elevations) being made equal in the two cells. To ensure these cells do not immediately dry out again, T_{Dam} is chosen as approximately $2T_{\mathrm{Dry}}$. This allows an element with bed elevation lower than the water free surface height in a neighbouring element to be wetted and thus allows the wetting front to advance down a slope. It is, however, a very crude approach and cannot be expected to reproduce accurately the hydraulics of a wetting front propagating downwards.

3. MODEL TESTING

Although ultimately the SFV model should be tested against validation data from real environmental flows, in this paper an initial verification is performed against a number of analytical solutions of the shallow water equations. Testing against analytical solutions [8, 32] allows the model's numerical performance to be validated independently of process representation issues. The accuracy of the model in generating solutions to the shallow water equations can thus be rigorously assessed.

3.1. Flow round a circular bend

The steady-state shallow water equations can be written in cylindrical polar co-ordinates (r, φ) in terms of the flow depth h and the radial and tangential velocity components u_r and u_φ [8]:

$$\frac{1}{r}\,\frac{\partial}{\partial r}(ru_r h) + \frac{1}{r}\,\frac{\partial}{\partial \varphi}(hu_\varphi) = 0 \tag{24}$$

$$\begin{bmatrix} u_r \dfrac{\partial u_r}{\partial r} - \dfrac{u_\varphi^2}{r} + \dfrac{u_\varphi}{r}\dfrac{\partial u_r}{\partial \varphi} + g\dfrac{\partial (h+z)}{\partial r} + \dfrac{gn^2|\mathbf{u}|u_r}{h^{4/3}} \\ u_r \dfrac{\partial u_\varphi}{\partial r} - u_\varphi \dfrac{\partial u_\varphi}{\partial \varphi} + \dfrac{u_\varphi u_r}{r} + \dfrac{g}{r}\dfrac{\partial (h+z)}{\partial \varphi} + \dfrac{gn^2|\mathbf{u}|u_\varphi}{h^{4/3}} \end{bmatrix} = \mathbf{0} \tag{25}$$

For uniform flow round a circular bend with uniform down reach slope, there is no dependence on φ, and flow is purely tangential, so the momentum equation becomes two coupled ordinary differential equations for $h(r)$ and $u_\varphi(r)$:

$$g\frac{\mathrm{d}h}{\mathrm{d}r} - \frac{u_\varphi^2}{r} = 0 \tag{26}$$

$$\frac{g}{r}\frac{\mathrm{d}z}{\mathrm{d}\varphi} + \frac{gn^2u_\varphi^2}{h^{4/3}} = 0 \tag{27}$$

Substituting (27) into (26) gives the following solutions for the water depth and velocities:

$$h = \left(C - \frac{1}{3gn^2r}\frac{\mathrm{d}z}{\mathrm{d}\varphi}\right)^{-3} \tag{28}$$

$$u_\varphi = \left(\frac{-h^{4/3}}{n^2r}\frac{\mathrm{d}z}{\mathrm{d}\varphi}\right) \tag{29}$$

The constant C is determined from the boundary conditions for the problem, in this case an inflow given at the upper end of the reach. Equation (26) can be interpreted as equating centripetal acceleration with the radial component of the free surface gradient, and (27) as a balance between free surface gradient in the downstream direction and friction terms. A comparison between (28) and (29) and the model results will therefore test the model's treatment of free surface gradient terms, friction and the cross stream component of the advection term.

The model was tested on a series of meshes (one of which is shown in Figure 5) of width 20 m and inner radius 50 m, with an inflow of 40 $\mathrm{m^3\,s^{-1}}$, Manning's n of 0.03 $\mathrm{m^{-1/3}\,s}$ and $\mathrm{d}z/\mathrm{d}\varphi = -0.06\,\mathrm{m\,rad^{-1}}$, giving a mean water depth of 1.46 m. These values were chosen as representative of a small meandering river flowing at bankful discharge. Model predicted flow depth and analytical solution for the mesh shown in Figure 5 is given in Figure 6, showing that the model predicts well the change in depth across the channel, but with a systematic overprediction of depth by $\sim$1 cm. The result was obtained using second-order spatial interpolation and an upwinding factor of 0.5. Figure 7 plots the mean error in predicted flow depth as a function of the angular deviation of the elements (element size divided by bend radius), along with results for the TELEMAC2-D finite element model [8], for both first- and second-order spatial interpolation. TELEMAC-2D is a finite element model solving the 2D shallow water equations on unstructured meshes of triangular elements, and is used here as a comparison because of the number of validation studies undertaken using the model. The SFV model developed here is a considerable improvement over TELEMAC2-D,

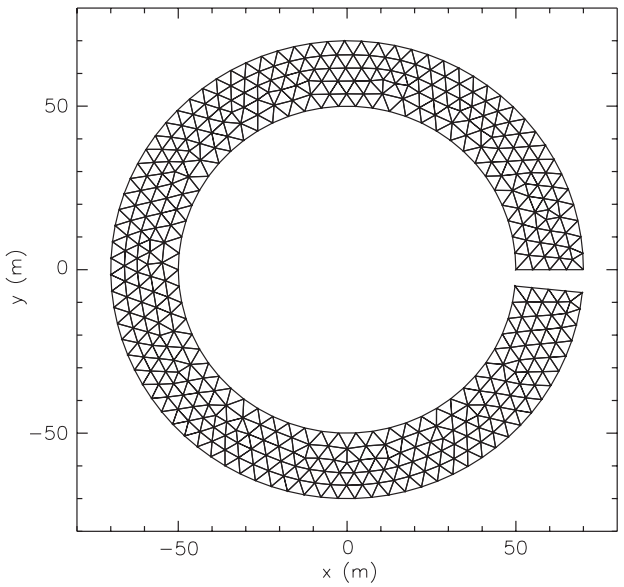

Figure 5. Example of computational mesh used to model flow round a circular bend.

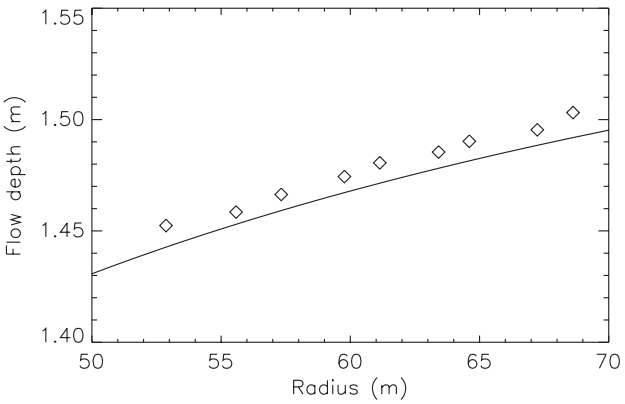

Figure 6. SFV predictions (◇) and analytical solution (solid line) for flow depth against radius for flow round a circular bend.

giving errors of <20 mm for all meshes. The results are also less strongly mesh dependent than for TELEMAC2-D. Some improvement is seen for the SFV model using second-order interpolation when compared with first-order.

3.2. Flow over an uneven bed

In this case we examine 1D flow over an uneven bed, and thus test the model's treatment of the free surface gradient and advection terms in the flow direction. For steady-state 1D flow,

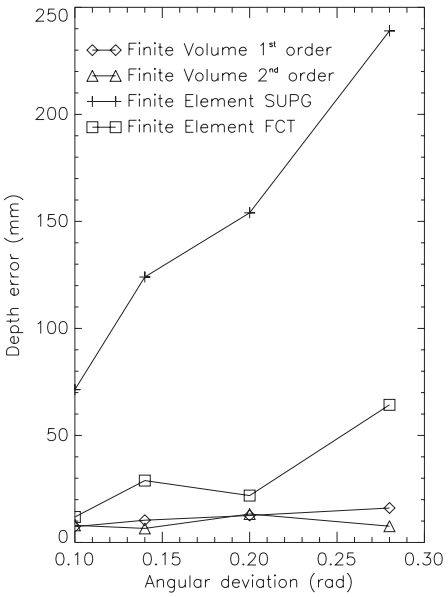

Figure 7. Error in SFV predictions when compared to analytical solutions as a function of element angular deviation for flow round a circular bend. The results for two versions of the TELEMAC-2D finite element model, using a streamline upwind Petrov–Galerkin (SUPG) scheme and a flux conservative transport (FCT) scheme, are also shown.

the continuity equation implies that uh $(=q)$ is constant along the reach, and if the friction term is set to zero the momentum equation can be written as

$$u\frac{\mathrm{d}u}{\mathrm{d}x}+g\frac{\mathrm{d}h}{\mathrm{d}x}+g\frac{\mathrm{d}z}{\mathrm{d}x}=0 \tag{30}$$

Integrating with respect to x gives a form of the Bernoulli equation:

$$\frac{q^2}{2h^2}+gh+gz+C=0 \tag{31}$$

The upstream boundary condition for the problem fixes q, and an imposed water depth at one point of the reach fixes the constant of integration C. Thus given appropriate boundary conditions, (31) can be used to find the analytical solution to the 1D shallow water equations at each point over a varying topography. The model is tested over a backward facing slope 40 m in length and a backward facing step, in a 500 m × 100 m rectangular domain discretized into 10 m elements. Boundary conditions were chosen to ensure a subcritical flow regime throughout the domain, with $q=1.0\,\mathrm{m^2\,s^{-1}}$ and depth at the downstream end fixed at 0.5 m, resulting in a depth upstream of 0.889 m. The results are given in Figure 8, for the 100 m in the region of the step, with points plotted for all elements across the domain. Errors increase from <30 mm for the 40 m slope up to <100 mm for the step, although in the latter case this error figure tends to understate model accuracy because of the discontinuity in the model free surface. Oscillations in the free surface are apparent downstream of the step, and model errors in the region of the step have propagated upstream to give a $\sim$5 mm error in the

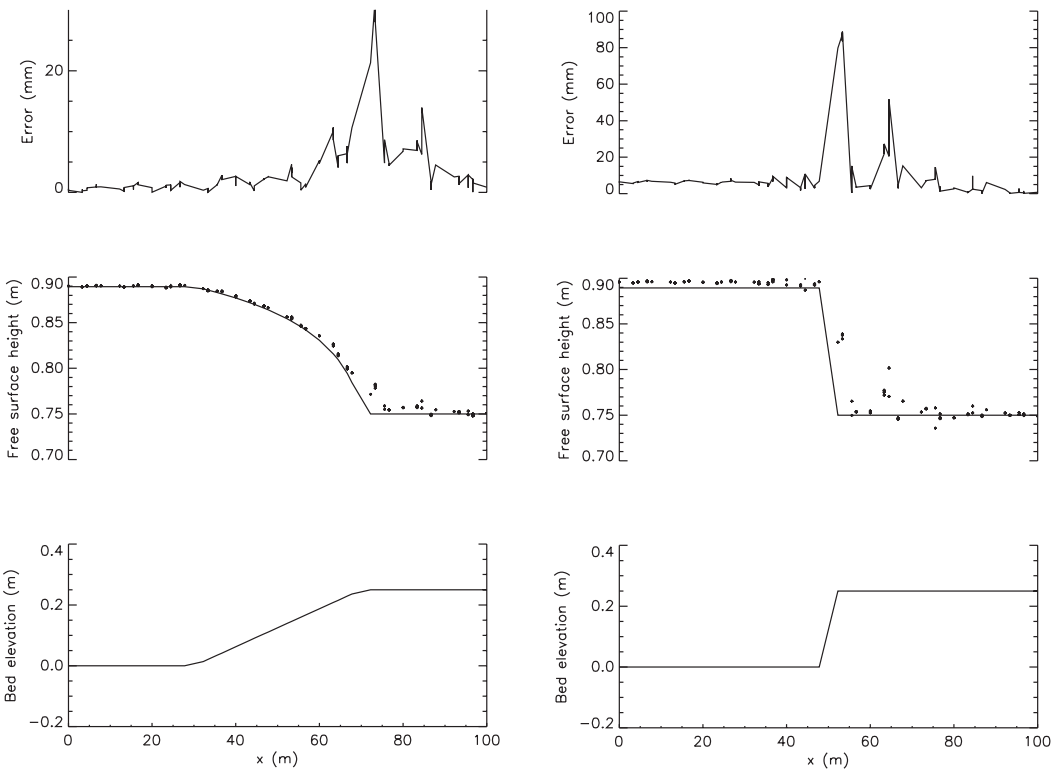

Figure 8. The results of SFV model and analytical solution for flow over a backward facing slope (lower left) and a backward facing step (lower right). The SFV solution (points) is compared to the analytical solution (solid line) in the middle plots. Absolute error is shown in the top plots (note different scales used for slope and step errors).

remainder of the reach. We would expect such persistent errors to be eliminated in a model with bed friction terms. The results are encouraging: even for step changes in topography, the model predicts the free surface reasonably in the region of the step, and predicts well the location and magnitude of the change in free surface elevation, and oscillations generated in the solution only affect a region 1 or 2 elements wide near the step. Figure 9 shows the results for an equivalent case, but with a Manning's n value of 0.01 $\mathrm{m}^{-1/3}$ s. The bed either side of the step has the slope required to give uniform flow conditions at the same depths as for the frictionless case. The results show errors in the region of the step of the same magnitude as for the frictionless case, but errors upstream of the step have been reduced to $\sim$1 mm.

3.3. *Floodwave propagation*

Although analytical solutions for time-varying problems are scarce, we can still check that the model reproduces the expected dynamic behaviour, and that the model produces mass conservative solutions. The model is therefore tested using a typical modelling scenario of a

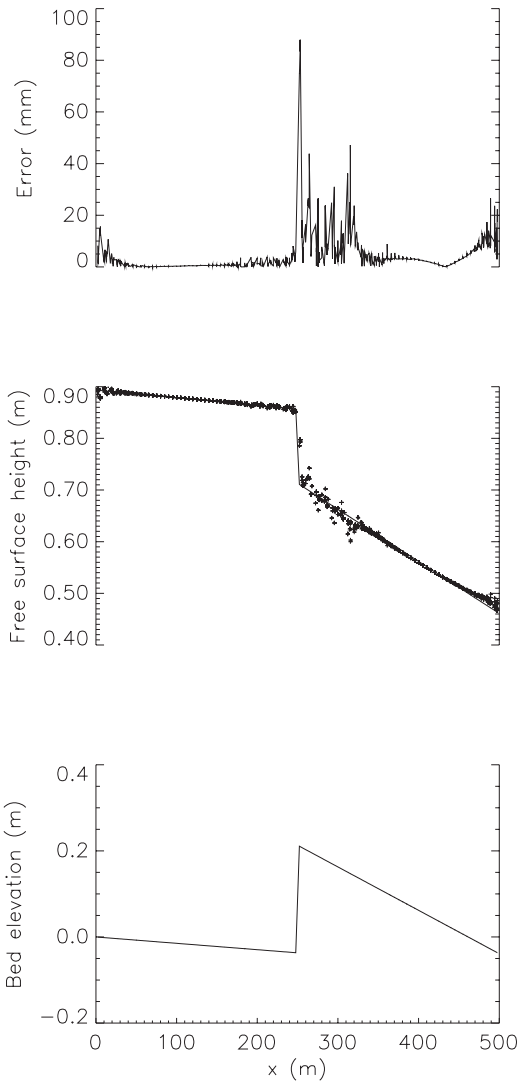

Figure 9. SFV model results for flow over a backward facing step with uniform bed slopes on either side, and friction terms included.

floodwave travelling down a meandering river reach, using five structured and unstructured meshes (Figure 10) as used in Reference [8]. Meshes 1–3 are structured with a regular pattern of triangular elements. Meshes 4 and 5 are unstructured. Meshes 2 and 3 use a curvature-dependent generation strategy designed to produce smaller elements in regions of high channel curvature. The reach is 1 km long, approximately 20 m wide, has a down reach slope of 10^{-3} and a Manning's n of $0.03\,\mathrm{m}^{-1/3}\,\mathrm{s}$. Model predictions of steady-state flow depth as a function of down reach distance are shown in Figure 11 for an inflow of $20\,\mathrm{m}^3\,\mathrm{s}^{-1}$. The results show good agreement ($\sim$2 cm) between solutions for all meshes except for number 2 (the lowest

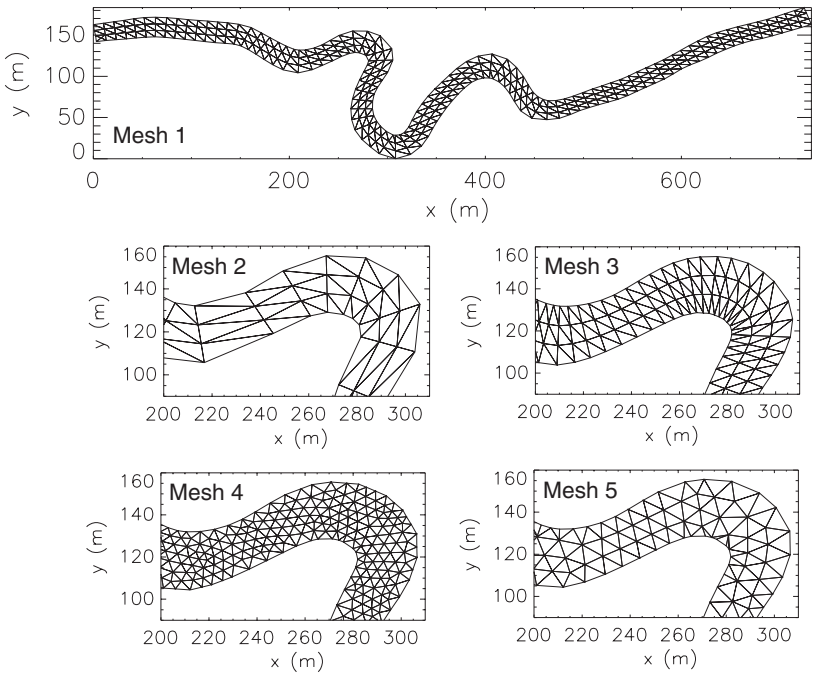

Figure 10. Computational meshes for the meandering river reach. Details of a bend section are shown for meshes 2–5.

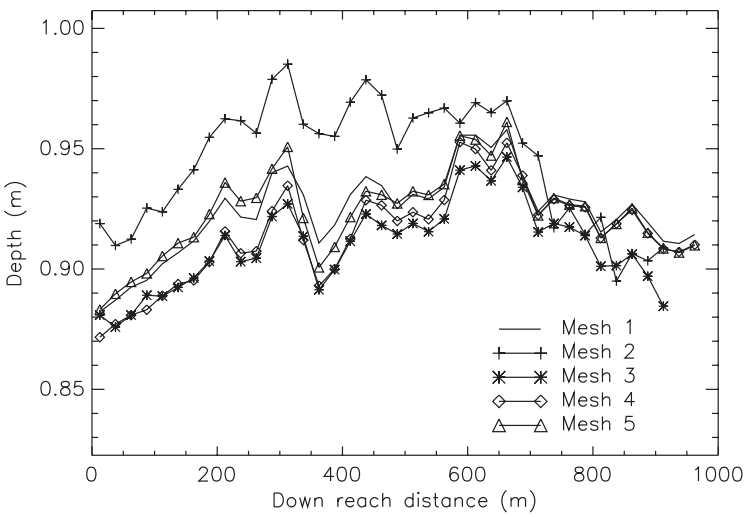

Figure 11. Flow depth (averaged across the channel) along the reach predicted by SFV for a steady-state inflow of 20 $m^3 s^{-1}$.

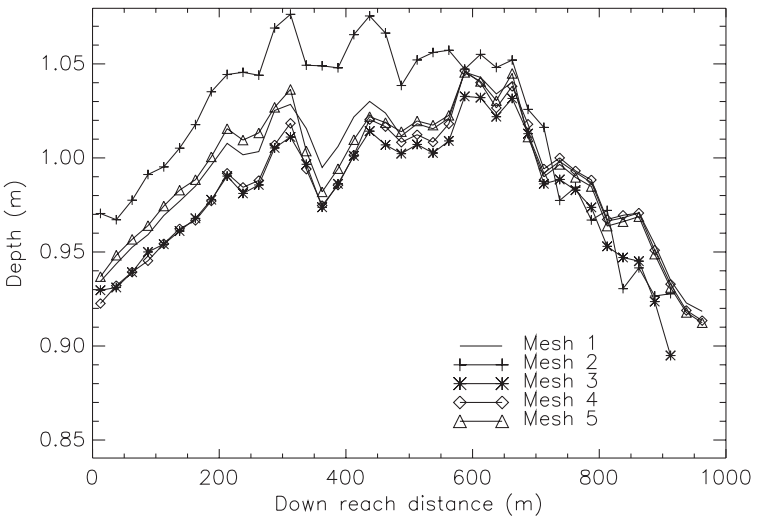

Figure 12. Flow depth along the reach predicted by SFV after the floodwave has propagated half-way down the reach. Note the change in vertical scale from Figure 10.

Table I. Mass balance errors for steady state and dynamic simulations for the 1 km river reach.

Mesh	Steady-state error ($m^3\ s^{-1}$)	Dynamic error ($m^3\ s^{-1}$)
1	-5.35×10^{-5}	-0.186
2	-2.38×10^{-5}	-0.237
3	-3.42×10^{-5}	-0.0937
4	-3.91×10^{-5}	-0.0460
5	-4.50×10^{-5}	-0.116

resolution), which predicts water levels $\sim$4 cm higher. The dynamic behaviour is tested using an input hydrograph that rises up to 30 $m^3\ s^{-1}$ and back down again over a time of 200 s. The predicted depths as the floodwave generated by the hydrograph has travelled half-way down the reach are shown in Figure 12. The same response as for the steady-state case is seen, with mesh two predicting water elevations to be $\sim$4 cm higher than the others. Table I gives the mass balance errors for the simulations, as the instantaneous maximum volume lost (negative values) or gained (positive values) per unit time:

$$Q_{\text{error}} = \frac{V^{n+1} - V^n}{\Delta t} - Q_{\text{in}} + Q_{\text{out}} \tag{32}$$

Q_{in} is the imposed flowrate, Q_{out} is the flow measured out of the model domain and V^n and V^{n+1} are the volumes of water in the domain at consecutive time steps. For steady-state simulations the mass balance error is calculated as the difference between the inflow and the outflow when the steady state is reached (i.e. volume is unchanging). The largest error occurs for mesh 2, but this value is still $<1\%$ of the total inflow. Compared with the results of a

TELEMAC-2D model of the same reach [8], SFV gives larger (but still acceptable) global mass balance errors, but the convergence between simulations using different meshes is far better ($\sim$2 cm compared with 10–15 cm for TELEMAC-2D).

3.4. *Wetting and drying*

SFVs treatment of moving boundary problems is first tested using a beach run up problem [32]. If a uniform bed slope S_0 is assumed, (1) and (2) can be written in a 1D form as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + gS_0 + g\frac{\partial h}{\partial x} + \frac{gn^2u|u|}{h^{4/3}} = 0 \tag{33}$$

$$\frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} = 0 \tag{34}$$

Purely advective solutions of the form $h(x - ct)$ can then be substituted, representing a wetting front propagating up the beach in response to an increasing water depth at the seaward boundary, while maintaining a constant profile. Equation (34) implies that the front propagation velocity c and the flow velocity u are identical, and [33] then becomes an ordinary differential equation

$$S_0 + \frac{\partial h}{\partial x} + \frac{n^2u^2}{h^{4/3}} = 0 \tag{35}$$

This equation can the be solved using relatively simple numerical techniques on a fine grid, such as a fourth-order Runge–Kutta scheme. This gives an accurate solution against which SFV can be tested. The solution can also be used to generate boundary conditions for SFV, which asymptotically tend towards a uniformly increasing water depth with time. The results from the Runge–Kutta scheme (implemented using a 5 m grid) and SFV are compared for three inundation rates and three SFV mesh sizes in Table II. The height errors (measured as RMS values at the centroids of the SFV elements) are comparable to those for TELEMAC-2D using a similar element masking wetting and drying algorithm, but with mass balance errors 1–2 orders of magnitude smaller for SFV. The results are also relatively insensitive

Table II. Performance of SFV model in reproducing wetting front profile for beach run up problem.

Simulation	Mesh size (m)	Time step (s)	Inundation rate ($m\,s^{-1}$)	Front velocity ($m\,s^{-1}$)	Height error (mm)	Mass balance error ($m^3\,s^{-1}$)
1	200	2.0	1×10^{-4}	0.1	1.19	0.01
2	100	2.0	1×10^{-4}	0.1	0.622	0.09
3	50	2.0	1×10^{-4}	0.1	0.496	5×10^{-3}
4	200	2.0	5×10^{-4}	0.5	8.07	0.26
5	100	2.0	5×10^{-4}	0.5	6.51	0.42
6	50	2.0	5×10^{-4}	0.5	6.20	0.36
7	200	0.5	1×10^{-3}	1.0	20.9	0.26
8	100	0.5	1×10^{-3}	1.0	19.8	1.71
9	50	0.25	1×10^{-3}	1.0	17.6	0.095

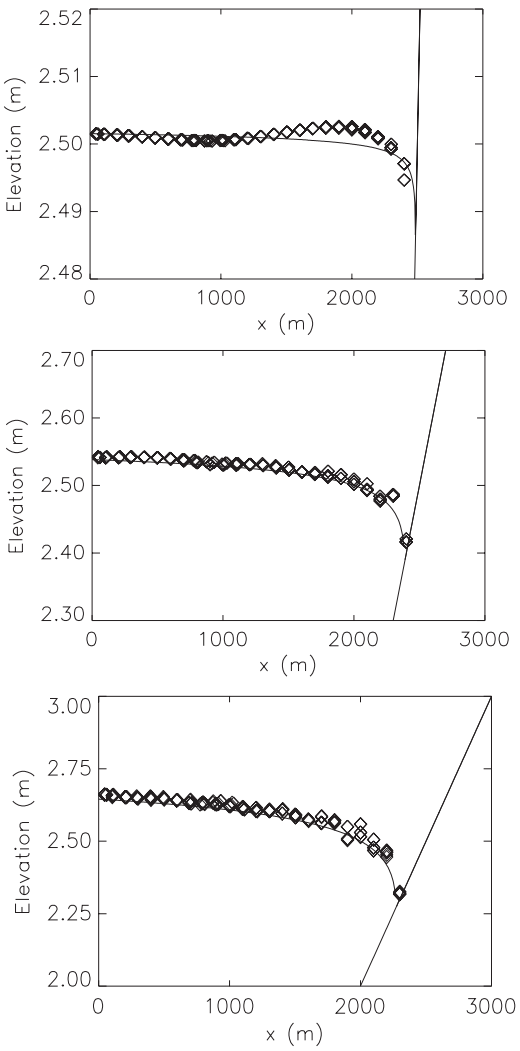

Figure 13. Water surface profiles predicted by SFV using the 200 m mesh (◇) and the Ruge–Kutta method (solid line) for the beach wetting problem for three inundation rates: 1×10^{-4} m^3 s^{-1} (top), 5×10^{-4} m^3 s^{-1} (middle) and 1×10^{-3} m^3 s^{-1} (bottom). Note the different vertical scales for the three inundation rates.

to the mesh resolution. Figure 13 shows free surface profiles for the SFV and Runge–Kutta techniques for the 200 m mesh for the three inundation rates. For the lowest rate, waves can be observed propagating in from the shoreline, possibly caused by the sudden motion in the shoreline as elements become wet. These waves are not evident for the faster inundation rates, where SFV captures well the draw down of the free surface towards the shoreline. While predicting the form of the water surface accurately, SFV has difficulty reproducing the correct near shoreline velocity vectors (Figure 14), which are strongly influenced by the mesh form

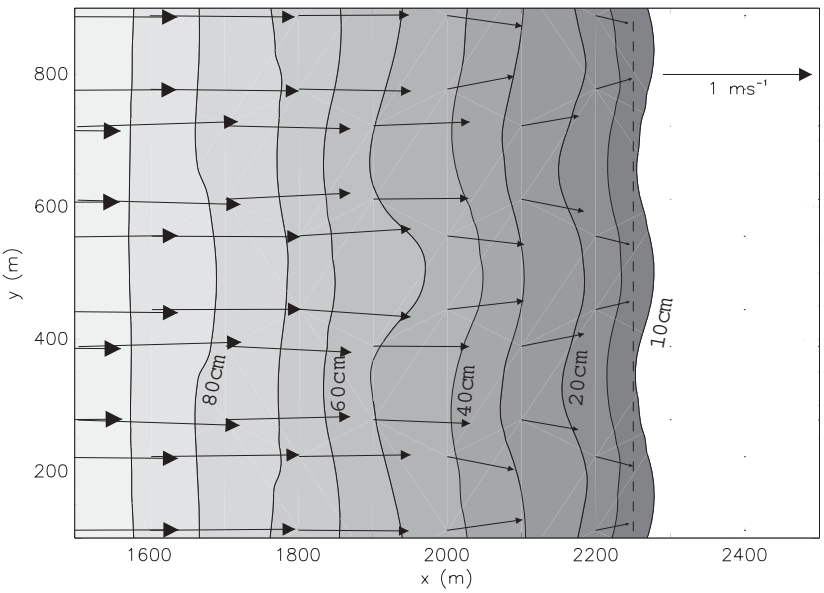

Figure 14. Flow vectors and flow depth predicted by SFV for inundation at a rate of $1\times10^{-3}\mathrm{m}^3\mathrm{s}^{-1}$ over the 200 m mesh. The 10 cm depth contour predicted by the Runge–Kutta method is also shown (dashed vertical line).

as elements become wet. The 10 cm depth contour predicted by the Runge–Kutta solution is also shown in Figure 14, and while its position along the beach is well predicted in the mean by SFV, the velocity variations produce some free surface variations that increase towards the shoreline. While no equivalent 1D advective solution exists for the drying case, SFV can still be tested in a qualitative fashion. Figure 15 shows the profiles for a drying simulation where the boundary condition was an imposed water surface dropping at a continuous rate of $1\times10^{-4}\,\mathrm{m\,s}^{-1}$. It shows the shoreline retreating smoothly down the beach with a draw up curve developing near the shoreline. Again variations in velocity vectors in the vicinity of the shoreline are observed as for the wetting case. The maximum mass balance error in this case was $0.013\,\mathrm{m}^3\,\mathrm{s}^{-1}$.

3.5. *Computational efficiency*

The computational efficiency of the SFV scheme was tested on a rectangular $1000\,\mathrm{m}\times500\,\mathrm{m}$ mesh with 50 m elements, and simulated uniform flow conditions with velocity $1.0\,\mathrm{m\,s}^{-1}$ and depth 1.0 m. Simulations were started with zero flow velocity and allowed to reach convergence, using a variety of time steps and implicitation factors corresponding to explicit, semi-implicit and fully implicit time development schemes, for first- and second-order spatial discretizations. Convergence was achieved in all cases after approximately 3700 s simulated time for first-order and 2900 s for second-order schemes. The results of this efficiency analysis are presented in Figure 16, showing the computational times required for the various spatial and temporal schemes and time steps. The results show that for the first-order scheme

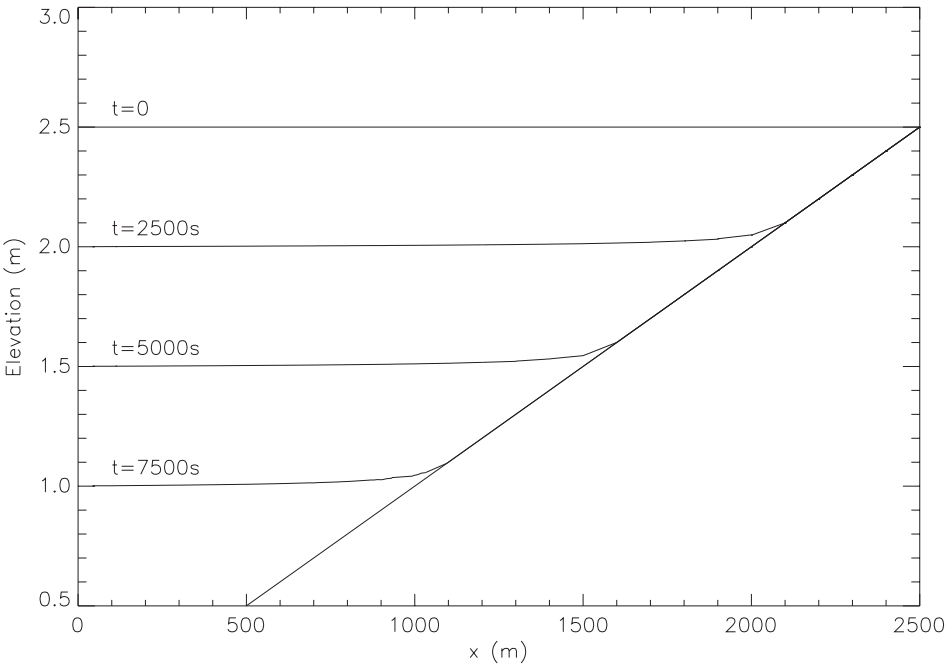

Figure 15. Free surface profiles predicted by SFV for beach drying at a rate of 1×10^{-4} m^3 s^{-1}.

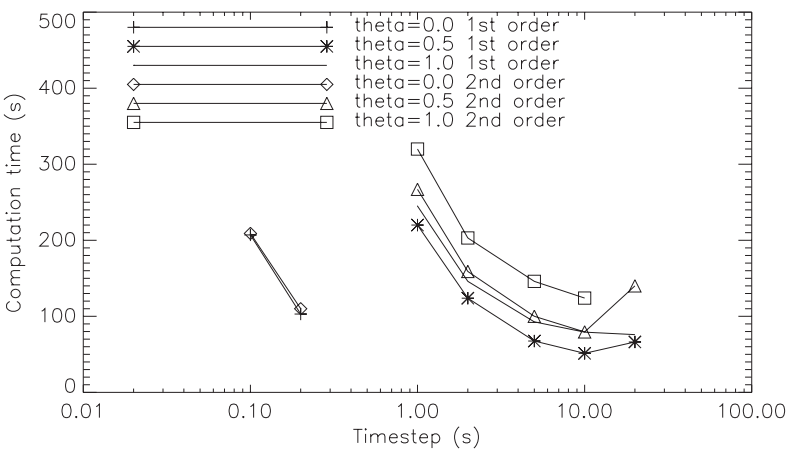

Figure 16. Computation times for SFV model using various implicit and explicit approaches and spatial integrations.

semi-implicit time development is most efficient, but that advantage is reduced for second-order spatial integration. Further reductions in computation times may be possible for the implicit schemes by refinement of the GMRES solver (e.g. use of preconditioning).

4. DISCUSSION

The comparison between SFV and the analytical solutions has shown that for these problems SFV is giving broadly satisfactory results. The mass balance errors produced by SFV are also generally encouraging, although no universally accepted criteria for acceptable mass balance errors exist. The greatest error encountered for the beach wetting problem, for example, is $1.71\ m^3\ s^{-1}$, which when taken in context represents only $\sim 0.1\%$ of the total flowrate into the reach at that time. Similarly for the flow along the river reach, errors are $< 1\%$ of the total flow into the reach. While such errors are certainly above the level expected if they were solely due to rounding error for example, they are at a level which is likely to be much less than the error in the boundary conditions dictating the flow into the model domain. This indicated that SFV's mass balance behaviour is acceptable. The errors are, however, probably greater than those due to inadequate process representation in the continuity equation, which neglects fluxes due to evaporation, infiltration, surface run off, etc. The river reach error for example is equivalent to a water loss of $\sim 40\ mm\ h^{-1}$.

SFV's performance in these tests seems also to be an improvement over the finite element method of TELEMAC-2D. Water levels for flow round the circular bend and predicted with greater accuracy by SFV, despite SFV only using a first-order accurate advection discretization. SFV's stability with respect to changes in mesh resolution is also an improvement over TELEMAC-2D which exhibits a strong mesh dependence. This is further demonstrated by the river reach test case where SFV exhibits closer convergence with respect to mesh size than TELEMAC-2D. The results for the beach wetting problem are also an improvement, with similar errors in the free surface height but with reduced mass balance errors shown by SFV. This is despite the less sophisticated solution representation used in SFV, where variables are assumed to be piecewise constant over each element, as opposed to the linear or quadratic representation typically used by finite element methods. This is however, partially offset by the smaller distances between element centroids, which are separated by 0.577 times the node spacing for an equilateral triangular mesh.

Further work required to incorporate features of other models into SFV is required. The inclusions of turbulence terms would be advantageous, although the inclusion and parameterization of turbulence schemes into 2D models is still a subject of debate [10]. A simple turbulent viscosity model has been built into a 2D finite volume methodology [21], but the calculation of second-order spatial gradients from a piecewise linear representation requires care. Higher order schemes such as the k–ε model [33] may also have to be used. The wetting and drying algorithm has also not been fully tested, and may need some refinement for modelling moving boundary flows over complex topography.

5. CONCLUSIONS

A simple finite volume model for 2D shallow water flow has been developed and tested against analytical solutions of the governing equations and the results of a finite element model. The model exhibits many desirable properties:

- Accurate prediction of the free surface of analytical solution for meandering channel flow.

- Good convergence properties with respect to computational mesh refinement.
- Accurate representation of free surface response to bed topography.
- Correctly handling wetting and drying fronts with only low levels of numerical oscillations near the shoreline.

The results indicate that this simple approach can be very effective in modelling sub-critical shallow water flows.

ACKNOWLEDGEMENTS

This work was supported by a U.K. Natural Environment Research Council fellowship. The author would like to thank Andrew Sleigh of the School of Civil Engineering at the University of Leeds for many helpful discussions.

REFERENCES

1. Alcrudo F, Garcia-Navarro P. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids* 1993; **16**(6):489–505.
2. Sleigh PA, Gaskell PH, Berzins M, Wright NG. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers and Fluids* 1998; **27**(4):479–508.
3. Beffa C, Connel RJ. Two-dimensional flood plain flow. I: Model description. *Journal of Hydrologic Engineering* 2001; **6**(5):397–405.
4. Beffa C, Connel RJ. Two-dimensional flood plain flow. II: Model validation. *Journal of Hydrologic Engineering* 2001; **6**(5):406–415.
5. Bermudez A, Dervieux A, Desideri J-A, Elena Vasquez M. Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:49–72.
6. Chippada S, Dawson CN, Dawson ML, Wheeler MF. A Godunov-type finite volume method for the system of shallow water equations. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:105–129.
7. Yu L. Two- and three-dimensional nested simulation by using FEM and FVA to analyse flows in an estuary. *Mathematical Computer and Modelling* 1998; **28**(11):115–134.
8. Horritt MS. Development of physically based meshes for two-dimensional models of meandering channel flow. *International Journal for Numerical Methods in Engineering* 2000; **47**:2019–2037.
9. Knight DW, Shiono K. River channel and floodplain hydraulics. In *Floodplain Processes*, Anderson MG, Walling DE, Bates PD (eds). Wiley, Chichester, 1996; 139–181.
10. Wilson CAME, Bates PD, Hervouet J-M. Comparison of turbulence models for stage-discharge rating curve prediction in reach-scale compound channel flows using two-dimensional finite element methods. *Journal of Hydrology* 2002; **257**(1–4):42–58.
11. Thornton CI, Abt SR, Morris CE, Fischenich JC. Calculating shear stress at channel-overbank interfaces in straight channels with vegetated floodplains. *Journal of Hydrologic Engineering* 2000; **126**(12):929–936.
12. Hagen SC, Horstman O, Bennet RJ. An unstructured mesh generation algorithm for shallow water modeling. *International Journal of Computational Fluid Dynamics* 2002; **16**(2):83–91.
13. Gee DM, Anderson MG, Baird L. Large scale floodplain modelling. *Earth Surface Processes and Landforms*. 1990; **15**:512–523.
14. Bates PD, Anderson MG. A two dimensional finite element model for river flood inundation. *Proceedings of the Royal Society of London* 1993; **440**(A):481–491.
15. Heniche M, Secretan Y, Boudreau P, Leclerc M. A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in Water Resources* 2000; **23**:359–372.
16. Makhanov SS, Vannakrairojn S, Vanderperre EJ. 2D Numerical model of flooding in east Bangkok. *Journal of Hydrologic Engineering* 1999; **125**(4):407–414.
17. Ip JTC, Lynch DR, Friedrichs CT. Simulation of estuarine flooding and dewatering with application to Great Bay, New Hampshire. *Estuarine Coastal and Shelf Science* 1998; **47**(2):119–141.
18. Sauveget P, David E, Soares CG. Modelling tidal currents on the coast of Portugal. *Coastal Engineering* 2000; **40**(4):393–409.
19. Bates PD, Hervouet J-M. A new method for moving boundary hydrodynamic problems in shallow water. *Proceedings of the Royal Society of London* 1999; **455**(A):3107–3128.
20. Roe PL. Approximate Riemann solvers, parameter vectors, and difference-schemes. *Journal of Computational Physics* 1981; **43**:357–372.

21. Anastasiou K, Chan CT. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997; **24**:122–1245.
22. Brufau P, Garcia Navarro P. Two-dimensional dam break flow simulation. *International Journal for Numerical Methods in Fluids* 2000; **33**:35–57.
23. Monthe LA, Benkhaldoun F, Elmahi I. Positivity preserving finite volume Roe schemes for transport-diffusion equations. *Computer Methods in Applied Mechanics and Engineering* 1999; **178**:215–232.
24. Osher S, Solomone F. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation* 1982; **38**:339–374.
25. Zhao DH, Shen HW, Lai JS, Tabios III GQ. Approximate Riemann solvers in FVM for 2D hydraulic shock wave modelling. *Journal of Hydrologic Engineering* 1994; **122**(12):692–702.
26. Harten A, Lax PD, van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
27. Hu K, Mingham CG, Causon DM. A bore-capturing finite volume method for open-channel flows. *International Journal for Numerical Methods in Fluids* 1998; **28**:1241–1261.
28. Hubbard ME, Baines MJ. Conservative multidimensional upwinding for the steady two-dimensional shallow water equations. *Journal of Computational Physics* 1997; **138**:419–448.
29. Wang J-W, Liu R-X. A comparative study of finite volume methods on unstructured meshes for simulation of 2D shallow water wave problems. *Mathematics and Computers in Simulation* 2000; **53**:171–184.
30. Bradford SF, Sanders BF. 2002. Finite-volume model for shallow-water flooding of arbitrary topography. *Journal of Hydrologic Engineering* 2002; **128**(3):289–298.
31. Saad Y, Schultz MH. GMRES—a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**(3):856–869.
32. Horritt MS. Evaluating wetting and drying algorithms for finite element models of shallow water flow. *International Journal for Numerical Methods in Engineering* 2002; **55**:832–851.
33. Stansby PK. Semi-implicit finite volume shallow-water flow and solute transport solver with k–ε turbulence model. *International Journal for Numerical Methods in Fluids* 1997; **25**:285–313.

