WEIGHTED IMPLICIT FINITE-VOLUME MODEL FOR OVERLAND FLOW

By A. M. Wasantha Lal,[1] Member, ASCE

ABSTRACT: A weighted implicit finite-volume model is developed to simulate two-dimensional diffusion flow in arbitrarily shaped areas. The model uses a mixture of unstructured triangles and quadrilaterals to discretize the domain, and a mixture of cell wall types to describe structures, levees, and flow functions that characterize two-dimensional flow. The implicit formulation makes the model stable and run faster with very large time steps. The sparse system of linear equations that results from the implicit formulation is solved by using iterative solvers based on various preconditioned conjugate gradient methods. The model was tested under a variety of conditions. The results were compared with results from known models applied to axisymmetric and other test problems that had known solutions. The model was applied successfully to the oxbow section of the Kissimmee River in Florida, and the results were compared with results from physical and numerical modeling studies. This analysis indicated that the circumcenter-based flow function for walls that is used in the model gives overall superior results in all the cases considered. Results of the numerical experiments showed that the use of weighted implicit methods and iterative solvers provide modelers with improved flexibility and control of the overall accuracy and the run time. The method is to be used as the solution method for the South Florida Regional Simulation Model.

INTRODUCTION

Simulation of overland flow is an important function of large-scale hydrologic models. Many such models, including the natural system model (NSM) and the south Florida water management model (SFWMM), which are used to simulate the hydrology of south Florida, are based on solving approximate forms of the St. Venant equations to simulate overland flow. An ideal model for the simulation of two-dimensional (2D) overland flow is expected to handle water bodies of arbitrary shape and may have to use a wide range of temporal and spatial features to meet accuracy requirements at different locations and times. Some of the historic developments related to this goal are described in the texts by Abbott (1979), Tan (1992), and Chaudhry (1993). The features that make models useful for practical applications include the ability to handle wetting and drying; the ability to simulate flow through structures such as weirs, gates, and culverts; and the ability to handle tributary and slough inflows.

The earliest 2D models to solve the St. Venant equations were based on various explicit finite-difference methods and rectangular grids. Liggett and Woolhiser (1967), Chow and Ben-Zvi (1973), and Katopodes and Strelkoff (1978) developed some of the early models. More recently, complete equation models have been developed that are capable of handling the inertia terms better and can produce better results for dam-break types of dynamic problems. Fennema and Chaudhry (1990) and Garcia and Kahawita (1986) have developed two such models. Finite-element and finite-volume methods are useful when the flow domain is arbitrary and the discretization is nonuniform. Fenner (1975) and Akanbi and Katopodes (1988) developed models based on the finite-element method, and Zhao et al. (1994) used a finite-volume method for solving the complete equations. Most of the complete equation models that use irregular grids require a long time to run and are inefficient to use in large-scale hydrologic applications, such as modeling of the Everglades, in which the inertia term is negligible. The challenge of maintaining both fine spatial resolution and low run times can be met by using diffusion flow models in which the inertia terms are neglected. In diffusion flow models one equation is solved for the water level, instead of the three coupled equations that form the St. Venant equations.

Ponce et al. (1978) established a theoretical range of applicability for diffusion flow models. Such models have been applied in the past by Xanthopoulos and Koutitas (1976) to simulate flood wave problems, by Akan and Yen (1981) to study channel confluence flow problems, and by Hromadka and Lai (1985) to study dam-failure problems. These studies showed that diffusion flow models can be used successfully to simulate a variety of natural flow conditions. Hromadka and Lai (1987) also used a 2D diffusion flow model to compare overland flow models. Diffusion flow models have been used successfully to simulate hydrologic conditions in the Everglades, using the NSM and the SFWMM models developed by the South Florida Water Management District (Fennema et al. 1994).

A finite-volume method is useful for south Florida because many of the postdrainage features in the area take the shape of polygons bounded by levees and canals. It satisfies strict mass balance because of conservative property. The basic idea behind the finite-volume method is to begin with the conservative form of the differential equation, integrate it over a finite volume, and use Gauss' theorem to convert results into surface integrals, which then can be discretized (Hirsch 1988). During the computation of these surface integrals along the cell walls, functions defining average wall fluxes are needed. Two types of functions are used in this paper, one that uses a line integral and one that uses the circumcenters (centers of the circumscribing circles) of triangles. In the case of structures or any other flow features, these wall functions are replaced with appropriate functions. When a cell-centered finite-volume method is used with rectangular grids, the finite-volume method collapses to a finite-difference method.

The ordinary differential equations resulting from the finite-volume formulation can be solved by using a weighted implicit method. The weighting factor that is used in many one-dimensional (1D) models such as DAMBRK (Fread 1973, 1988) provides control over accuracy and stability, and the weighting also makes it possible to produce solutions even under stiff conditions. The final solution of the finite-volume method is the solution of a sparse system of linear equations at every time step. The availability of a variety of sparse solver methods and packages has made it possible to exercise control over the run time and accuracy.

Both direct and iterative methods are available to solve

[1]Lead Civ. Engr., South Florida Water Mgmt. Dist., 3301 Gun Club Rd., West Palm Beach, FL 33406.

Note. Discussion open until February 1, 1999. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on June 20, 1997. This paper is part of the ***Journal of Hydraulic Engineering***, Vol. 124, No. 9, September, 1998. ©ASCE, ISSN 0733-9429/98/0009-0941–0950/$8.00 + $.50 per page. Paper No. 16051.

sparse systems. Iterative methods, such as the preconditioned conjugate gradient (CG) method, are less susceptible to roundoff error, and they are more efficient for large problems (Aziz and Settari 1979). Some of the public domain sparse solvers available through the Internet include SLAP (Seager 1988), Templates (Barrett et al. 1993), and IML++ (Dongarra et al. 1995). Numerous preconditioners are used with sparse solvers to speed convergence and sometimes to make the solution feasible. When flow conditions are nearly steady because of negligible disturbances from rainfall and other events, iterative solvers need very few iterations. This feature can make the current model run extremely fast except during unsteady events.

Hydrologic models applied to the south Florida landscape are expected to simulate both large-scale flow features in the Everglades and small-scale flow features in urban areas. They are expected to be capable of both long- and short-term simulations with relatively short run times. This paper describes the formulation, numerical testing, numerical error analysis, and the successful application of the model to a portion of the Kissimmee River. A number of additional tests were conducted to study the variation in numerical error with spatial and temporal discretizations. Results demonstrate the fast performance of the model when compared with explicit models. The results also are useful in selecting the spatial and temporal discretization for future applications of the model to other areas in south Florida (Lai and Belnap 1997). Some results shown at low resolutions give additional information about the behavior of numerical errors in the model output.

GOVERNING EQUATIONS

Overland flow is described by the depth-averaged flow equations commonly referred to as St. Venant equations. These equations consist of a continuity equation and momentum equations. The 2D continuity equation for shallow water flow is

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} - \text{RF} + \text{IN} + \text{ET} + q_{ea} = 0 \quad (1)$$

where u and v = velocities in x- and y-directions; h = water depth in units L; RF = rainfall intensity; IN = infiltration rate; ET = evapotranspiration rate, all in units L/T; and q_{ea} = volume rate of overland flow entering or leaving canals, measured per unit cell area per unit time. The momentum equations used in the x- and y-directions are

$$\frac{\partial (hu)}{\partial t} + \frac{\partial (u^2 h)}{\partial x} + \frac{\partial (uvh)}{\partial y} + hg \frac{\partial (h + z)}{\partial x} + ghs_{fx} = 0 \quad (2)$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial (uvh)}{\partial x} + \frac{\partial (v^2 h)}{\partial y} + hg \frac{\partial (h + z)}{\partial y} + ghs_{fy} = 0 \quad (3)$$

where s_{fx} and s_{fy} = components of friction slopes in x- and y-directions. The momentum equations can be combined with the continuity equation without the source term to produce the following vector momentum equation

$$\frac{\partial \mathbf{V}}{\partial t} + \nabla \left(\frac{1}{2} V^2 + gH \right) + g\mathbf{s}_f + \mathbf{V} \times \boldsymbol{\omega} = 0 \quad (4)$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{V}$; $\mathbf{V} = u\mathbf{i} + v\mathbf{j}$ = velocity vector; $\mathbf{s}_f$ = friction slope vector; $H = h + z$ = water level above the datum; and z = bottom elevation above datum. The steps in obtaining the equation are presented by Panton (1984). Eq. (4) can be integrated along a streamline to obtain the commonly used energy equation. The first term in (4), which is the local acceleration term, and the second term, which is the convective acceleration term, are responsible for inertia effects. The first term is neglected in slowly varying flow to obtain diffusion flow equations. If flow is irrotational, $\boldsymbol{\omega} = 0$ and (4) reduces to

$$\nabla E = -\mathbf{s}_f \quad (5)$$

which also can be written in terms of the x and y components as $\partial E/\partial x = -s_{fx}$ and $\partial E/\partial x = -s_{fy}$ with $E = h + z + V^2/(2g) = H + V^2/(2g)$ being the energy head above the datum. Eq. (5) without the velocity head in E is used normally as the foundation of diffusion flow formulations, in which the water level H is used instead of the energy head E (Hromadka et al. 1987). Even if all of the equations that follow are expressed in terms of H, it can be shown that H in these equations can be replaced with E to give the necessary equations for conditions under which the velocity heads are important. This simple conversion is possible in slowly varying flow if $\partial/\partial t(V^2/2g)$ is small. Use of E instead of H helps to recover some of the lost inertia effects in slowly varying diffusion flow at converging and diverging boundaries. Unfortunately, diffusion flow models using the velocity head generate small oscillations in unsteady flow problems (Strelkoff et al. 1977), and it becomes necessary to use H instead of E for such problems.

The friction slope s_f in (5) is computed using an equation for wetlands (Kadlec and Knight 1996) or a general form of the Manning equation written as $V = (1/n_b)h^{\gamma}s_f^{\lambda}$ in which n_b = Manning coefficient when $\gamma = 2/3$ and $\lambda = 1/2$; $V = \sqrt{v^2 + u^2}$ = magnitude of the velocity vector. In diffusion flow $\mathbf{s}_f = \mathbf{s}_n$ is assumed, in which $\mathbf{s}_n$ = slope of the water surface (or the energy surface when E is used) computed as $\sqrt{(\partial H/\partial x)^2 + (\partial H/\partial_y)^2}$. Akan and Yen (1981) and Hromadka et al. (1987) used the following equation to compute u and v:

$$u = -\frac{K}{h}\frac{\partial H}{\partial x}, \quad v = -\frac{K}{h}\frac{\partial H}{\partial y} \quad (6)$$

The K can be expressed for the Manning equation in general form as

$$K = \frac{1}{n_b} h^{\gamma+1} \quad \text{for} \quad \lambda \geq 1 \quad \text{and} \quad |\mathbf{s}_n| > \delta_s \quad (7)$$

$$K = K_0 \quad \text{for} \quad \lambda < 1 \quad \text{and} \quad |\mathbf{s}_n| \leq \delta_s \quad (8)$$

Term $K_0 = h^{\gamma+1}/(n_b \delta_s^{1-\lambda})$ provides continuity in function K and gives a smoother flow profile for some problems than $K_0 = 0$ used by Hromadka (1985). Depth $h = 0$ for dry cells. The δ_s is used to bound K within finite limits; $\delta_s \approx 10^{-10}$ is used in the study for test cases in single precision. The $\lambda \approx 1$ gives laminar-like flow; K is useful in linearizing and simplifying the diffusion flow equation. The continuity equation in (1) can be expressed, using (6), as

$$\frac{\partial H}{\partial t} = \frac{\partial}{\partial x} K \frac{\partial H}{\partial x} + \frac{\partial}{\partial y} K \frac{\partial H}{\partial y} + \mathbf{S} \quad (9)$$

where $\mathbf{S} = \text{RF} - \text{IN} - \text{ET} - q_{ea}$ is the source term. When the velocity head is included, H in (9) is replaced with E, as explained earlier. The equation can be solved for both surface flow and saturated ground-water flow using many of the methods used to solve parabolic equations.

Finite-Volume Method

In the finite-volume method (1) is expressed in the following integral form over an arbitrary control volume cv:

$$\frac{\partial}{\partial t} \int_{cv} H \, dv + \int_{cv} \left[\frac{\partial}{\partial x} (hu) + \frac{\partial}{\partial y} (hv) - \mathbf{S} \right] dv = 0 \quad (10)$$

where dv = volume of element cv. The overall cv can be subdivided into cells. The Gauss divergence theorem can be used to simplify the second volume integral term of (10) and convert

it to a surface integral (Hirsch 1988). Eq. (10) for all the finite-volume cells can be written in vector form as

$$\Delta\mathbf{A}\cdot\frac{d\mathbf{H}}{dt} = \mathbf{Q}(\mathbf{H}) + \mathbf{S} \tag{11}$$

where $\mathbf{H} = [H_1, H_2, \ldots H_m \ldots H_{nc}]^T$ = a vector containing the average heads in all the cells; $\mathbf{S}$ = the source term in vector form; $\Delta\mathbf{A}$ = a diagonal matrix whose element $\Delta\mathbf{A}(m, m)$ is equal to the cell area ΔA_m in the case of a cell m; and $\mathbf{Q}$ and $\mathbf{S}$ = net inflows and source term to cells. The net inflow rate to a cell m is given by

$$Q_m(H) = \sum_{r=1}^{ns} (\bar{\mathbf{F}}\cdot\mathbf{n})_r \Delta l_r \tag{12}$$

where Δl_r = length of the side r of the ns sided polygon; $\mathbf{n} = n_x\mathbf{i} + n_y\mathbf{j}$ = unit outward normal vector for the face r of the polygon; $\bar{\mathbf{F}}$ = average flux rate across the wall per unit length defined as $hu\mathbf{i} + hv\mathbf{j}$, which also is equal to $-K\nabla H$ for free surface diffusion flow or ground-water flow. Two alternative methods are used in the model to compute $\bar{\mathbf{F}}$ for overland flow. They are the line-integral-based method suggested by Hirsch (1988) and the circumcenter-based method suggested by Cordes and Putti (1996). In the case of flow over structures and levees, $Q_m(H)$ is computed using the appropriate structure equations instead of the foregoing two methods. In the current cell-centered finite-volume approach, H, ET, RF, and IN are defined as cell average values.

Line-Integral-Based Method for Computing Wall Flux

This method can be used with both triangular and quadrilateral cells. With the use of this method, the approximate flux $\bar{\mathbf{F}}_r$ for a wall r in (12) is computed by using fluxes at the nodes defining the wall. In Fig. 1

$$\bar{\mathbf{F}} = 0.5(\hat{\mathbf{F}}_j + \hat{\mathbf{F}}_k) \tag{13}$$

where $\hat{\mathbf{F}}_j$ and $\hat{\mathbf{F}}_k$ = fluxes at the nodes j and k computed using $-K\nabla H$. The ∇H is computed using an integral equation around the nodes (Hirsch 1988) such that

$$\int_v \nabla H\, da = \oint_s H\mathbf{n}\, dl \tag{14}$$

and dl = length of the sides of the polygon, referred to as the "shadow polygon," with cell centroids at vertices. With the use of (14), the flux $\hat{\mathbf{F}}_j$ for a node j can be expressed as

$$\hat{\mathbf{F}}_j = -K_j(\nabla\mathbf{H})_j = -\frac{K_j}{2\Delta\hat{A}_j}\left[-\sum_{p=1}^{np} H_p(y_{p+1} - y_{p-1})\mathbf{i} + \sum_{p=1}^{np} H_p(x_{p+1} - x_{p-1})\mathbf{j}\right] \tag{15}$$

where $p = 1, 2, \ldots, np$ = cell numbers around the node j forming the vertices of the shadow polygon; and x_p, y_p = coordinates of these vertices. In the equation, x_0, y_0 at $p = 1$ must be replaced by x_{np}, y_{np}, and x_{np+1}, y_{np+1} at $p = np$ must be replaced by x_1, y_1 to complete the integration correctly. Areas of the shadow polygons $\Delta\hat{A}_j$ are computed by using a similar line integration

$$2\Delta\hat{A}_j = \sum_{p=1}^{np} x_p(y_{p+1} - y_{p-1}) \tag{16}$$

The K_j are computed using (7) and (8). The nodal values of n_b and h in the equations are obtained by a weighted averaging of the values of surrounding cells. The respective cell areas are used as weights. The line integrals are computed counterclockwise as positive.

In the use of the weighted implicit implementation, $\mathbf{Q}(\mathbf{H}) = [Q_1, Q_2 \ldots Q_{nc}]^T$ of (11) is linearized as $\mathbf{M}\cdot\mathbf{H}$. The matrix $\mathbf{M}$ contains information about the connectivity among cells, geometry, and the roughness. The matrix $\mathbf{M}$ is assembled by computing the flow rates across all of the walls using (12) and adding or subtracting appropriate volumes from the cells. Consider the volume lost by donor m, crossing wall r defined by nodes j and k. Eqs. (12) and (13) and the line integral around node j obtained using (15) makes the following modification to $\mathbf{M}$:

$$M_{m,p} \rightarrow M_{m,p} - \frac{K_j\Delta l_r}{4\Delta\hat{A}_j}[-n_{xr}(y_{p+1} - y_{p-1}) + n_{yr}(x_{p+1} - x_{p-1})], \quad p = 1, \ldots, np \tag{17}$$

where n_{xr}, n_{yr} = components of $\mathbf{n}$ for wall r; and Δl_r = length of wall r. A similar expression is needed for node k. Flow into the receiver cell n also requires two similar expressions with negative signs placed on (n_{xr}, n_{yr}).

Circumcenter-Based Method for Computing Wall Flux

Cordes and Putti (1996) showed the equivalence of a low-order mixed finite-element method based on RT0 elements (Raviart and Thomas 1977) with a finite-volume method for triangles under certain conditions. Because of the equivalence, it is possible to use an expression derived for the mixed finite-element method to compute flow rates for the finite-volume

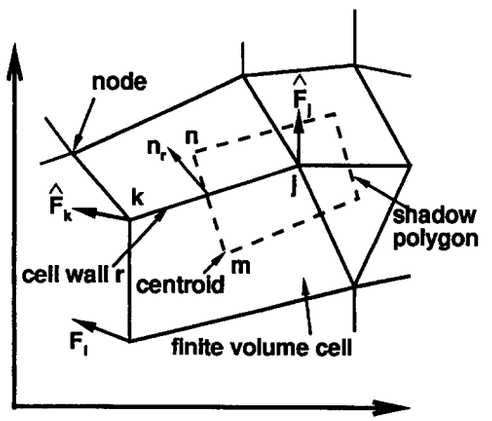

FIG. 1. Diagram Showing Definition of Variables Used in Line Integral Method

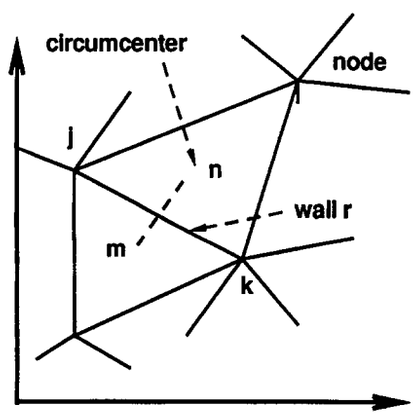

FIG. 2. Diagram Showing Definition of Variables Used in Circumcenter Method

method. In the equivalent finite-volume method, water levels at circumcenters are used in the computation of flow across walls. In the mixed finite-element method, water levels in triangles are assumed to vary linearly, and the water level at the centroid is the average water level. Using Fig. 2 as the definition sketch, $(\hat{\mathbf{F}} \cdot n)_r$ for wall r in (12) is computed as

$$(\hat{\mathbf{F}} \cdot n)_r = \Delta l_r K_r \frac{H_m - H_n}{\Delta d_{mn}} \tag{18}$$

where Δd_{mn} = distance between circumcenters of triangles m and n; and H_m, H_n = heads at the circumcenters. The K_r is computed using (7) or (8). The depth and the bed roughness needed to compute K_r are obtained by weighted averaging the depth and bed roughness of cells m and n. The S_n is computed using

$$S_n = \sqrt{\frac{(\hat{H}_j - \hat{H}_k)^2}{\Delta l_r^2} + \frac{(H_m - H_n)^2}{\Delta d_{mn}^2}} \tag{19}$$

where $\hat{H}_j$ and $\hat{H}_k$ = heads at nodes j and k, computed as weighted averages of surrounding heads. The cell areas are used as weights in the averaging. In the semi-implicit formulation the computation of flow from cell n to m involves the modification of the following matrix element as it receives water in cell m:

$$M_{m,n} \rightarrow M_{m,n} + \frac{K_r \Delta l_r}{\Delta d_{mn}}, \quad M_{m,m} \rightarrow M_{m,m} - \frac{K_r \Delta l_r}{\Delta d_{mn}} \tag{20}$$

Elements $M_{n,m}$, $M_{n,n}$ are modified similarly because of water losses from the donor cell n. The circumcenter-based method can be used only with acute-angled triangles. When this method is used with obtuse-angled triangles, the circumcenter falls outside the triangle and the numerical error tends to be large. With rectangles the method becomes equivalent to the finite-difference method.

The average water velocity in a cell is computed by using the following vector basis function developed for RT0 mixed elements of Raviart and Thomas (1977) and used by Cordes and Putti (1996):

$$\mathbf{v} = \frac{1}{2Ah}\left[Q_{s1}\begin{pmatrix} x - \hat{x}_1 \\ y - \hat{y}_1 \end{pmatrix} + Q_{s2}\begin{pmatrix} x - \hat{x}_2 \\ y - \hat{y}_2 \end{pmatrix} + Q_{s3}\begin{pmatrix} x - \hat{x}_3 \\ y - \hat{y}_3 \end{pmatrix}\right] = -K\nabla H \tag{21}$$

where Q_{s1}, Q_{s2}, Q_{s3} = discharge rates across cell walls $s1$, $s2$, and $s3$ counting outward as positive; $(\hat{x}_i, \hat{y}_i)$ = the coordinates of the nodes; and (x, y) = coordinates of any point, including the circumcenter in the current case at which the head is computed. In the case of right-angled triangles, Cordes and Putti (1996) showed that the mixed finite-element method is equivalent to a finite-difference method.

Flow Through Structures and Levees

When the model is used to simulate structure flows, the specific cell walls are replaced with structure-type walls, and flow rates of $Q_s(H)$ are used in (12) instead of $\mathbf{F} \cdot \mathbf{n}$ to compute structure flows. Linearization of structure flow equations can be done either prior to the run using regression methods or during the run using data from previous calls to the routine. $Q_s(H)$ is computed as a function of adjacent water levels, gate openings, and other physical parameters. Assuming that the variation of Q_s versus ΔH ($\Delta H = H_m - H_n$) is linear during two consecutive time steps, a structure equation can be developed using the information collected during the time steps p and $p - 1$ as

$$Q_s(\Delta H) = Q_s^p + K_s(\Delta H - \Delta H^p) \quad \text{for} \quad \Delta H^p \neq \Delta H^{p-1} \tag{22a}$$

$$Q_s(\Delta H) = Q_s^p \quad \text{otherwise} \tag{22b}$$

where $K_s = (Q_s^p - Q_s^{p-1})/(\Delta H^p - \Delta H^{p-1})$; and p = the time step count. If only the information at time step p is used, (22) reduces to $Q_s(\Delta H) = K_s \Delta H$ and the right side of the system of equations need not be modified. The introduction of a structure between cells m and n modifies $\mathbf{M}$ as $M_{m,n} \rightarrow M_{m,n} + K_s$, $M_{m,m} \rightarrow M_{m,m} - K_s$, $M_{n,m} \rightarrow M_{n,m} + K_s$, and $M_{n,n} \rightarrow M_{n,n} - K_s$ as in (20). In the computations it was assumed that the head loss caused by bed friction is negligible when compared with head loss across structures. If iterations are carried out within a time step, the linearization will not introduce errors in the solution. Because rapid flow variations are not expected in diffusion flow, the linearization gives good results even for structures having nonlinear flow relations.

When there is a structure or a levee-type cell wall, the 2D flows in adjacent cells are affected and become more nearly 1D. The following equation, based on the Manning equation, is applied between cells across a wall under this condition:

$$Q_{1d} = K_n \Delta H = \frac{h^{\gamma+1} \Delta l_r}{n_b \Delta d}\left(\frac{\Delta H}{\Delta d}\right)^{\lambda-1} \Delta H \tag{23}$$

Here, n_b and h are averaged between cells; and Δd = distance between the cell centroids. Centroids are used to represent cell locations in restricted spaces or closer to structures and dry cell where free 2D flow cannot be assumed and slope S_n of the water surface profile cannot be determined accurately. For these cells K_n is computed by assuming that the water surface slope S_n in the Manning equation is approximately equal to $\Delta H/\Delta d$.

Boundary Conditions

One boundary condition is needed at each boundary with diffusion flow. Specified head and specified flow are the most commonly used types. The no-flow type boundary is implemented simply by making $\bar{\mathbf{F}} = \mathbf{0}$ in (12). The matrix $\mathbf{M}$ needs no modification under no-flow conditions. In the case of a known inflow rate Q_I into a cell i through the boundary or caused by pumping activity, row i of source term $\mathbf{S}$ in (11) must be modified as

$$\mathbf{S}_i \rightarrow \mathbf{S}_i + Q_I \tag{24}$$

Source term quantities such as rainfall, ET, and infiltration are summed similarly for cell i.

If the flow domain is connected to an external reservoir as the boundary condition and if the reservoir water level is H_0 the equation for flow rate into the domain Q_0 is linearized as $Q_0 = K_0(H_0 - H_i)$, in which K_0 is similar to the structure constant K_s in (22) and H_0 and H_i are water levels of the water body and the cell. The modifications for matrix $\mathbf{M}$ and vector $\mathbf{S}$ are $M_{i,i} \rightarrow M_{i,i} - K_0$ and $\mathbf{S}_i \rightarrow \mathbf{S}_i + K_0 H_0$. Implementation of head boundary conditions is explained later.

Formulation of Weighted Implicit Method

The ordinary differential equations in (11) derived using the finite-volume method are solved by using the following weighted finite-difference formulation:

$$\Delta A_i H_i^{n+1} = \Delta A_i H_i^n + \Delta t[\alpha Q_i^{n+1} + (1 - \alpha) Q_i^n] + \Delta t[\alpha \mathbf{S}_i^{n+1} + (1 - \alpha)\mathbf{S}_i^n] \tag{25}$$

where H_i^n = average surface water level in cell i at time step n; α = time weighting factor; and $\alpha = 0$ and 1 for explicit and implicit problems. With the use of linearization, (25) can be expressed as the following system of linear equations:

$$[\Delta\mathbf{A} - \alpha\Delta t\mathbf{M}^{n+1}]\cdot\Delta\mathbf{H} = \Delta t[\mathbf{M}^n]\cdot\mathbf{H}^n + \Delta t(1-\alpha)[\mathbf{M}^n - \mathbf{M}^{n+1}]\cdot\mathbf{H}^n + \Delta t[\alpha\mathbf{S}^{n+1} + (1-\alpha)\mathbf{S}^n] \quad (26)$$

Here, $\mathbf{Q}^n = \mathbf{M}^n\cdot\mathbf{H}^n$. The solution $\Delta\mathbf{H}$ is used to update the heads using $\mathbf{H}^{n+1} = \mathbf{H}^n + \Delta\mathbf{H}$. The matrix $\mathbf{P} = [\Delta\mathbf{A} - \alpha\Delta t\mathbf{M}^{n+1}]$ is so far symmetric. In many gradually varying problems $\mathbf{M}^{n+1}$ is replaced with $\mathbf{M}^n$ to simplify (26) (Akan and Yen 1981). Test runs show that this is a useful procedure for many problems. If this assumption is not made then $\mathbf{M}^{n+1}$ must be updated by using an iterative procedure within the time step, by first computing $\Delta\mathbf{H}$ using (26) with the most recent estimates of $\mathbf{M}^{n+1}$ and next updating $\mathbf{H}^{n+1}$. Iterations are continued similarly by updating $\mathbf{M}^{n+1}$ and using (26) until convergence. Examples used in the paper need only 2–4 iterations for the convergence of the water level up to four significant digits. This type of iteration was not used in the current application.

Imposition of a head boundary condition to a cell i as $H_i = H_B$ is carried out by reconfiguring row i of $\mathbf{P}$. The entire row i is modified by using

$$P_{i,j} = 0 \quad \text{for} \quad j = 1, 2, \ldots nc, \quad j \neq i \quad (27a)$$

$$P_{i,j} = 1 \quad \text{for} \quad j = 1, 2, \ldots nc, \quad j = i \quad (27b)$$

$$\mathbf{S}_i = H_B - H_i^n \quad (27c)$$

Matrix $\mathbf{P}$ is sparse for large problems. The element density is less than 1% for a 1,000 cell discretization. When $\alpha = 0$, $\Delta\mathbf{H}$ in (26) can be computed by using a simple matrix multiplication. The $\alpha = 0.5$ gives higher accuracy as in the case of Crank-Nicholson-type schemes. With rectangular grids the finite-volume method gives the finite-difference solution.

Solution of Linear Equations

The number of equations in the system of linear equations in (26) is equal to the number of cells nc. If the cells are nonuniform and the physical properties are nonhomogeneous, the problem may become stiff and the matrix $\Delta\mathbf{A} - \alpha\Delta t\mathbf{M}$ may become ill conditioned. However, many fast efficient iterative sparse solvers that can handle ill-conditioned matrices recently have become available. The current model was tested with the SLAP solver (Seager 1988) and the PetSc solver (Smith et al. 1995). Both solvers use iterative CG methods and preconditioners. Preconditioners are useful in improving the convergence rate and the solvability. Without preconditioning, the number of iterations increases with the condition number. The condition number of a matrix is the ratio of the largest and smallest eigenvalues. If the system of equations becomes difficult to solve with the chosen sparse solver, Δt can be reduced until $\mathbf{A} - \alpha\Delta t\mathbf{M}$ becomes well conditioned. The need to rerun the code because of nonconvergence can sometimes be avoided by reusing $\mathbf{M}$ with a smaller Δt.

Active research is under way to develop faster sparse solvers. A feature available with faster packages gives one the ability to solve equations at each time step as a sequential process and incrementally improve the solution by starting from the solution for the previous time step. Without such methods the same or nearly the same equations still may have to be solved repeatedly at steady or near-steady conditions, wasting computer resources. Many of the new features in solvers can make the model run much faster during such events by carrying out the minimum required updating from one time step to the next and using only a few iterations, depending on the extent of transient flow activities.

NUMERICAL TESTS

The model was tested for accuracy by applying it to a number of test problems with known solutions. The first test was used to check the ability of the finite-volume method to solve diffusion equations accurately. The second test was carried out with 2D diffusion-type overland flow. The remaining tests were designed to carry out a numerical error and stability analysis.

Test 1

A ground-water example from Wang (1982) was used for the first test. In the test a pumping well was positioned at the center of a 4,000 × 4,000 m square confined aquifer having a constant transmissivity (K) of 300 m^2/day and a storage coefficient of 0.002. A uniform initial water level of 10 m and a

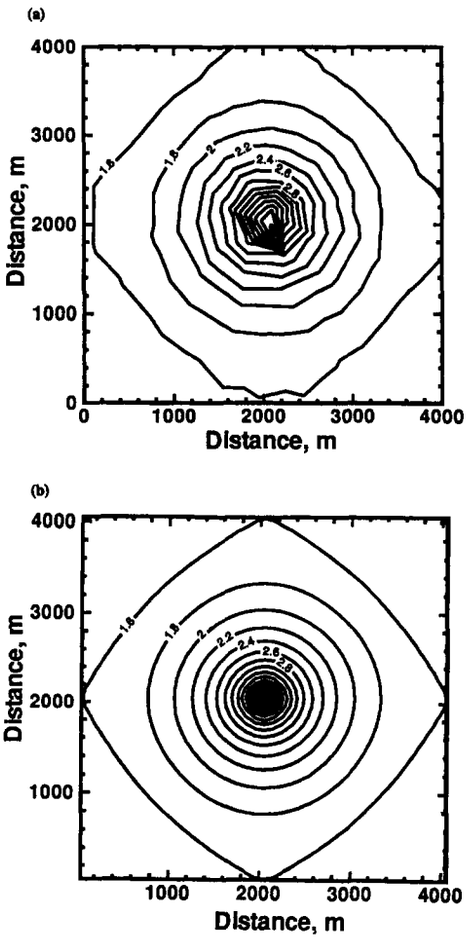

FIG. 3. Drawdown Contours Obtained Using: (a) Finite-Volume Model; (b) MODFLOW Model

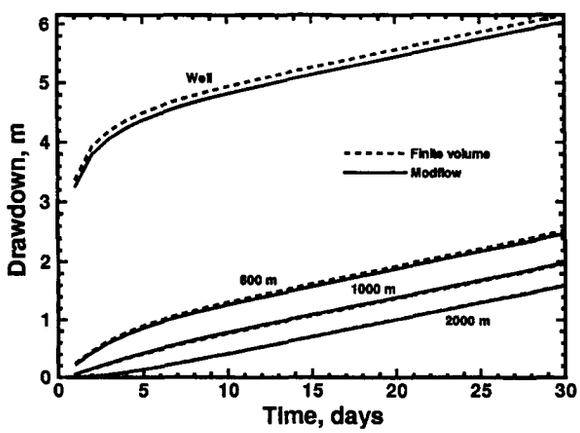

FIG. 4. Variation of Drawdown with Time at Different Distances

constant pumping rate of 2,000 m^3/s were assumed. The triangular discretization used with the model is the same as that shown later in Fig. 5 with 238 cells, except that the linear dimensions are scaled down to fit the area into the 4,000 × 4,000 m square. The MODFLOW model (McDonald and Harbaugh 1984) was set up to simulate the same flow conditions using a 40 × 40 square grid with 1,600 cells. Fig. 3(a) shows the water level contours at the end of 30 days, obtained by using the circumcenter-based finite-volume method. Fig. 3(b) shows the same contours obtained by using the MODFLOW model. Drawdown curves at a number of monitoring points are shown in Fig. 4. The finite-volume method using the line-integral-based flow function failed to produce convex water level contours near the well, and the results are not shown. The test shows that the circumcenter-based finite-volume method with only 238 cells can produce relatively accurate solutions. The test also shows that the circumcenter-based method gives better results than the line-integral-based method for locally converging flow.

Test 2

An axisymmetric overland flow problem was used in the second test. The flow characteristics of this test are somewhat similar to the flow characteristics of the Everglades. The test bed has dimensions 161 × 161 km (100 × 100 miles) and a flat bottom. The initial condition is

$$H = \left[0.4575 + 0.1525 \cos\left(\frac{\pi r}{r_{\max}}\right)\right] \text{m} \quad \text{for} \quad r \leq r_{\max} \tag{28}$$

$$H = 0.305 \text{ m} \quad \text{otherwise} \tag{29}$$

where r = distance from the domain center; and $r_{\max}$ = 32,188 m. The Manning roughness is assumed as 1.0; RF, IN, and ET are neglected. An axisymmetric diffusion flow model was developed based on the following axisymmetric continuity equation to obtain an extremely accurate solution for the problem using a fine resolution

$$\frac{\partial(hr)}{\partial t} + \frac{\partial(uhr)}{\partial r} = 0 \tag{30}$$

This solution was used in computing small numerical errors in the finite-volume model under different resolutions. A model, similar to the 1D model by Akan and Yen (1981) after a few modifications, was used to solve (30) accurately. The test was a 12-day simulation of the water level using both the axisymmetric model and the finite-volume model. In the test, Δr = 80.47 m and Δt = 1 min were used with the axisymmetric model to obtain the water level in the problem that was accurate enough to compute numerical errors in other models. The error at the center was used for comparison purposes because the error is largest at this point. The water level computed accurately at the center is 0.442105 m. The expected circular shape of the solution also was used to test accuracy of the finite-volume models.

The finite-volume model using the circumcenter-based approach was used with discretizations of different refinements to recreate the results of the axisymmetric model. The results, obtained using a discretization of 238 cells and 135 nodes and a time step of 3 h, are shown in Fig. 5. The SLAP 2.0 sparse solver package (Seager 1988) was used to solve the linear equations, and convergence was assumed when the largest change in the solution vector $\varepsilon_\infty < 0.3 \times 10^{-4}$ m. Other parameter values used were α = 0.5 and $\delta_s = 1.0 \times 10^{-10}$ [in (7) and (8)]. The figure shows the grid used and the contour plot of water levels after 12 days. The water level at the center of the circular patch and at cells at radial distances of r = 11,885 m and r = 31,000 m was monitored during the simulation. Fig.

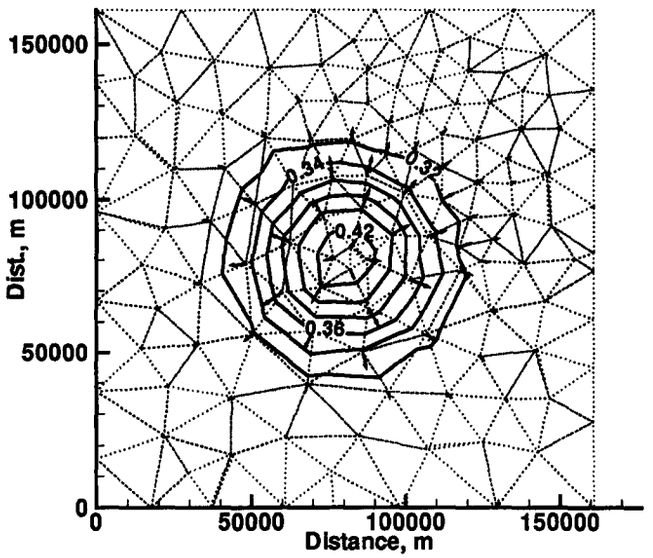

FIG. 5. Contour Plot of Water Levels in Axisymmetric Test Problem

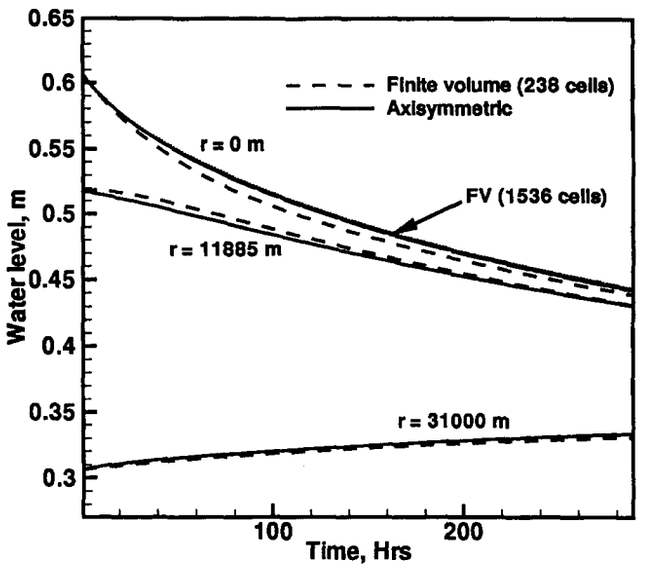

FIG. 6. Variation of Water Level with Time in Axisymmetric Test Problem

6 shows the general agreement of water levels at all the monitoring points, using both the axisymmetric model and the finite-volume model. Fig. 6 also shows the solution at r = 0 obtained using a finite-volume model running with a time step of 3 h and a higher resolution obtained using 1,536 cells. As seen in the figure, the finite-volume solution very closely matches with the axisymmetric solution at this high resolution.

Numerical Error and Stability

The accuracy of results obtained from a numerical model depends on the spatial and temporal discretizations. If a model is used to simulate flow features of a certain wavelength, the resolution of the mesh should be sufficient to capture that wavelength. A description of the variation of the numerical error with the spatial and temporal resolutions is provided by Lal (1998). To understand the behavior of the numerical error in the current finite-volume model, triangular meshes of different levels of discretization were used in the simulation of the flow pattern used in the previous test case. The Groundwater Modeling System software package (*Department* 1995)

TABLE 1. Solutions of Test Problems Using Various Discretizations

Test (1)	No. of elements (2)	No. of nodes (3)	CPU (s) (4)	No. of iterations (5)	Δx (m) (6)	Δt (s) (7)	h_{end} (m) (8)	π/ϕ (9)	β (10)	ε% (11)
1	116	69	2.4	18	14,939	51,840	0.4488	2.15	0.016	1.09
2	116	69	8.8	12	14,939	10,368	0.4484	2.15	0.003	1.03
3	116	69	16.4	11	14,939	5,184	0.4484	2.15	0.002	1.02
4	376	209	6.0	40	8,298	207,360	0.4450	3.88	0.212	0.48
5	376	209	25.1	19	8,298	20,736	0.4446	3.88	0.021	0.40
6	376	209	43.6	17	8,298	10,368	0.4444	3.88	0.011	0.38
7	376	209	78.8	13	8,298	5,184	0.4444	3.88	0.005	0.37
8	1,536	809	60.1	104	4,105	518,400	0.4540	7.84	2.166	1.96
9	1,536	809	75.3	78	4,105	207,360	0.4449	7.84	0.866	0.48
10	1,536	809	98.3	67	4,105	103,680	0.4450	7.84	0.433	0.48
11	1,536	809	258.0	35	4,105	20,736	0.4439	7.84	0.087	0.29
12	1,536	809	436.0	27	4,105	10,368	0.4437	7.84	0.043	0.27
0	238	135	27.7	1	10,429	5,184	0.4390	—	0.50	0.49

Note: Results of test 0 with nonhomogeneous cells are shown in Figs. 5 and 6. CPU = central processing unit time.

was used to generate meshes for this test. An estimate of the numerical error was obtained for comparison purposes by presenting the numerical error at $r = 0$ after 12 days as a percentage of the depth at $t = 0$. Numerical error was computed by using the previously mentioned axisymmetric solution as the true solution because it has an error term much smaller than the error studied. Table 1 shows a summary of test results for the center, obtained by using circumcenter-based methods. Run times shown are for a SUN Sparc 20 [speed 90 MHz, 4.1 Mflops/s measured with the linpack benchmark test (Dongarra 1993)]. The iterations shown are the iterations inside the SLAP 2.0 solver indicating the computational effort. In the table Δx was computed as $\sqrt{\Delta A_c}$ in which ΔA_c is the average area of a triangular cell. The ϕ is obtained as $k\Delta x$, in which k is the wave number of the water surface profile simulated in the model = 2π/(wavelength). Term π/ϕ gives an estimate of the spatial resolution, measured as the average number of spatial divisions within half the wavelength of a sinusoidal water surface profile. The β is the nondimensional time step size, which is based on the analysis of Lal (1998)

$$\beta = \frac{h^{5/3}}{n_b\sqrt{S_n}}\frac{\Delta t}{\Delta x^2} \tag{31}$$

The $\beta < 0.25$ for explicit finite-difference methods. Test 0 corresponds to the test shown in Figs. 5 and 6 for 238 cells. Results of test 12 with 1,536 cells also is shown in Fig. 6. Table I shows that the solution of the finite-volume model approaches the axisymmetric solution as the spatial and temporal resolutions both get finer. This is true when the model is using the line-integral-based method too. Table 1 also shows that the run time decreases and the number of iterations per time step increases when the time step is increased.

A test was conducted to check the stability of the model under explicit conditions ($\alpha = 0$). Experimentation with different time steps showed that Δt at the points of incipient instability of the tests was approximately 52, 4.3, and 3.5 h, respectively, with 116, 376, and 1,536 cell configurations listed in Table 1. These time steps correspond to approximate β values of 0.06, 0.02, and 0.05, respectively. Incipient instability was assumed when dynamic oscillations were visible at the center of the solution. These results confirm, e.g., that tests 8–11 in Table 1, obtained for $\alpha = 0.5$, would have been unstable under explicit conditions. The approximate stability limit $\beta \approx 0.05$ is useful in selecting the time step for explicit model runs. Nonlinear instability was not studied during the test.

Numerical tests were conducted to determine the convergence behavior of the finite-volume code and the influence of δ_s in (8) on the performance of the code. Tests showed that the number of iterations increased when δ_s was decreased to very low values, because some of the K values in the matrix became very large (Lal and Belnap 1997), and the matrix became more unconditional and ultimately unsolvable as a result. The solution errors at the center after 12 h were 1, 21, and 88 mm as δ_s was changed to 10^{-6}, 10^{-5}, and 10^{-4}, respectively. A large δ_s causes the model to use (8) instead of (7) more often. The $\delta_s = 10^{-10}$ was used in the axisymmetric flow test, and $\delta_s = 10^{-4}$ was used in the Kissimmee study that is explained later.

Different sparse solver options in the SLAP 2.0 package were tested while running test 0 referred to in Table 1. The purpose of the test was to investigate the performance of different solvers and preconditions. In the SLAP 2.0 package the incomplete LU decomposition with CG solver, incomplete LU biconjugate gradient solver, and the incomplete LU biconjugate gradient solver with LU decomposition were reliable and used the least number of iterations. The last option was used in the test. The number of solver iterations changed with the solver type and δ_s, which affects the condition number of the matrix. With large time steps the SLAP 2.0 solver converged only when large α values are used. The recently developed PetSc solver (Smith et al. 1995) was found to be much more reliable and fast for larger problems.

Application to Kissimmee River

The model was applied to the experimental area near weir no. 2 of the Kissimmee River basin, Florida, using the same discretization and the bed roughness used by Zhao et al. (1994). In the application by Zhao et al. the unsteady flow model RBFVM-2D was used over the test area shown in Fig. 7, which is approximately 1,402 × 1,036 m. In the figure a flood canal passes from the north to the south (left to right in the figure), and a one-notch weir is located near the upstream end near C1 to divert part of the flow into the river oxbow. The Manning coefficients of the floodplain, main channel, and the river oxbow are 0.03, 0.025, and 0.04, respectively. The number of nodes and cells in the mixed grid used by the RBFVM-2D model and the line-integral-based finite-volume model are 347 and 327, respectively. The same numbers in the case of the circumcenter-based method are 347 and 634, respectively. For the circumcenter-based method the quadrilaterals were divided into triangles. The results of the problem for an inflow of 221 m^3/s at the upstream boundary and a stage of 13.57 m at the downstream boundary are shown in Fig. 7, after running the model until a reasonable steady state is reached. The results were obtained after including the velocity head $V^2/(2g)$ in (5). When the same simulation was repeated with the omission of the velocity head, the water level at C1

dropped by 1 cm. Water levels at other locations remained practically unchanged. Fig. 7 shows contours of water levels and the water level monitoring points. The elliptical patch of contours in the figure shows a small dry area. Fig. 8 shows the velocity vectors drawn at the circumcenters using (21). The apparent overlap of arrows in the plot is caused by the near right-angled triangles in the grid, which make the circumcenters nearly overlap. Fig. 9 shows the results of the same test obtained using the line-integral-based method.

Comparison of water levels and water velocities in Table 2 shows that the water levels obtained with the current model agree with the physical model results and the RBFVM-2D model results at many locations. However, velocities at O2, representing a narrow canal segment of the oxbow, that were obtained by using diffusion flow models did not agree with other velocities. Comparison of the circumcenter-based method with the line-integral-based method shows that both methods produced similar flow patterns in the Kissimmee application, unlike in the test cases with a locally convergent or divergent flow field in which the line-integral-based method produced unacceptable local results. This occurred because the averaged $\bar{F}$ in (13) does not provide a very accurate estimate of discharges across walls in acute-angled triangles. Certain velocities near the boundary are not shown in Table 2 because line integrals could not be computed with this method without a closed path of integration.

With the Kissimmee application it also was found that the line-integral-based method required approximately 50 iterations when using 20-s time steps and the SLAP CG method using LU decomposition preconditioner. The circumcenter method required approximately 200 iterations for the same case. The run time for the current model is a small fraction of the run time of explicit models such as RBFVM-2D requiring

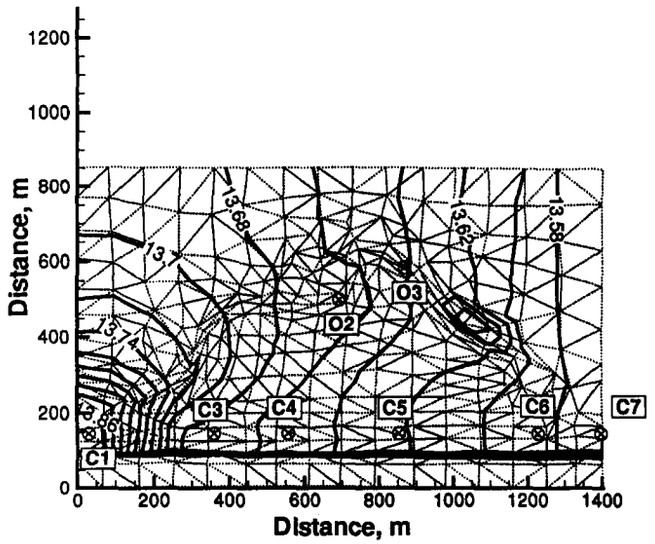

FIG. 7. Contour Plot of Water Levels in Kissimmee River, Obtained Using Circumcenter Method

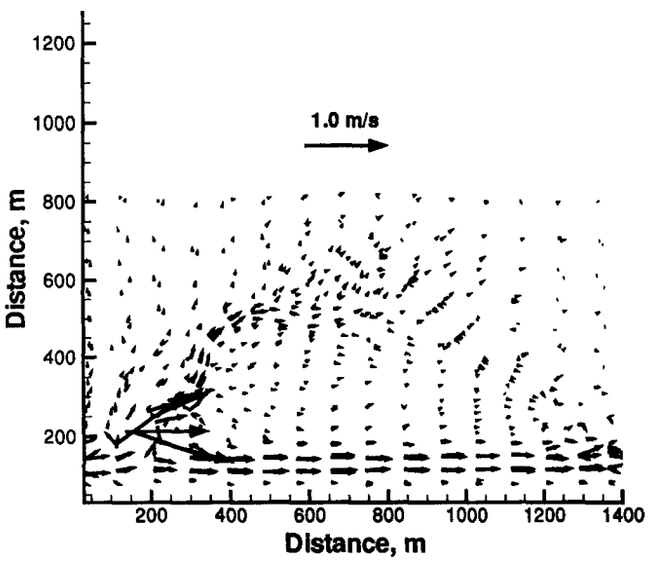

FIG. 8. Vector Plot of Water Velocities in Kissimmee River, Obtained Using Circumcenter-Based Walls

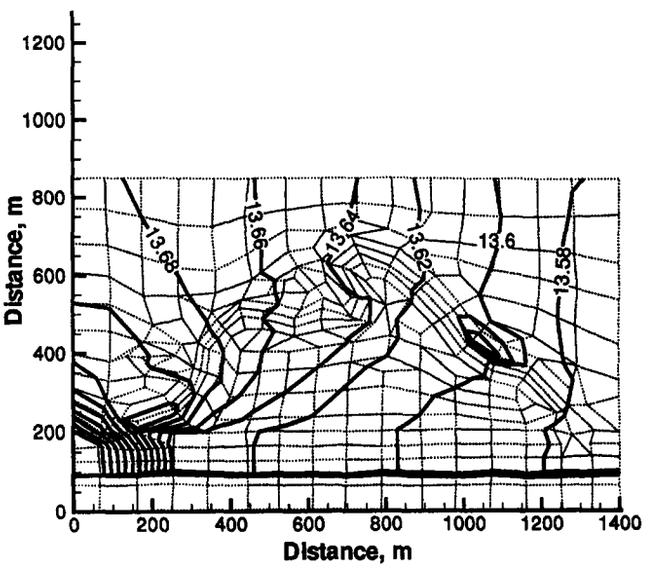

FIG. 9. Contour Plot of Water Levels in Kissimmee River, Obtained Using Line-Integral-Based Walls

TABLE 2. Comparison of Physical Model Results with Results of Finite-Volume Models Using Circumcenter-Based Walls and Line-Integral-Based Walls

Gauge (1)	Physical Model		RBFVM-2D Model		Circumcircle Method		Line Integral Method	
	Velocity (m/s) (2)	Stage (m) (3)	Velocity (m/s) (4)	Stage (m) (5)	Velocity (m/s) (6)	Stage (m) (7)	Velocity (m/s) (8)	Stage (m) (9)
C1	0.30	13.87	0.29	13.78	0.24	13.87	—	13.85
C3	0.23	13.57	0.21	13.60	0.26	13.66	—	13.62
C4	0.23	13.57	0.25	13.60	0.25	13.61	0.28	13.61
C5	0.23	13.57	0.29	13.57	0.25	13.60	0.29	13.59
C6	0.23	13.57	0.31	13.58	0.27	13.57	0.27	13.58
C7	0.29	13.57	0.33	13.69	0.21	13.57	—	13.57
O1	0.85	13.67	0.67	13.69	0.98	13.77	0.70	13.73
O2	0.49	13.67	0.44	13.64	0.06	13.60	0.14	13.64

Note: Results of the finite RBFVM-2D model by Zhao et al. (1994) also are shown.

1–2 s time steps. The PetSc solver (Smith et al. 1995) with a new C++ version of the the current model can reduce the number of iterations to less than five with even larger time steps and make the model run much faster. With the development of better and faster external sparse solvers using parallel processing and other methods, large-scale application of the model to south Florida continues to become less expensive, just with the upgrading of the solver.

SUMMARY AND CONCLUSIONS

An implicit finite-volume model was developed to simulate diffusion flow across arbitrarily shaped landscapes. Tests were conducted to verify the model results by comparing them with results from the MODFLOW model and an axisymmetric model. The model also was applied to a variety of test problems, using a range of spatial and temporal discretizations to study the behavior of numerical errors. Results show that numerical errors tend to become smaller with finer discretizations, thus confirming the numerical consistency condition. The explicit option ($\alpha = 0.0$) showed incipient instability when the nondimensional time step β exceeds approximately 0.05. The implicit option was stable for large values of β. Results show that, by selecting a spatial resolution (π/ϕ) of more than approximately three divisions per half sine wave, numerical errors for the test problems can be reduced to less than 1%.

The model used different wall types to represent structure flows, no flows, and 2D flows. Flow across 2D walls was computed by using a line-integral-based method and a circumcenter-based method. Results show that the circumcenter-based method produced better results under all the conditions tested, and that the line-integral-based methods produced local errors when used with triangular discretizations to simulate locally convergent or divergent flow patterns. The line-integral-based method becomes the choice when polygons, not triangles, are used in the discretization. This method also needed fewer iterations inside the solver when used with test problems. Application of both methods to the Kissimmee River shows that the results agree with the results of the physical model and the RBFVM-2D model. The same application showed that, although the RBFVM-2D model needed 1–2 s time steps, the current model could be run faster with time steps over 10 times as large, even with older solvers, and many more times faster with modern solvers.

The structure of the current finite-volume model allows new wall flow function types to be added to the existing circumcenter and line integral types and new structure types to be added in the same way. This feature is useful for future extensions of the model into more complicated areas of south Florida and the Everglades. Increasingly powerful sparse solvers can continue to speed computations in the future and make it possible to simulate flows with much finer spatial resolutions and larger time steps otherwise possible, as demonstrated in the examples.

ACKNOWLEDGMENTS

The writer thanks Mark Belnap, Randy Van Zee, and Jayantha Obeysekara for providing valuable ideas during different stages of model development. Review of the manuscript by Todd Tisdale, Joel VanArmen, Steve Lin, Brion Lehar, Sashi Nair, Mark Wilsnack, Ken Tarboton, and other members of the Hydrologic Systems Modeling group of the South Florida Water Management District was extremely useful.

APPENDIX I. REFERENCES

Abbott, M. B. (1979). *Computational hydraulics*. Ashgate Publishing Co., Brookfield, Vt.

Akan, A. O., and Yen, B. C. (1981). "Diffusion-wave flood routing in channel networks." *J. Hydr. Div.*, ASCE, 107(6), 719–731.

Akanbi, A. A., and Katopodes, N. D. (1988). "Models for flood propagation on initially dry land." *J. Hydr. Engrg.*, ASCE, 114(7), 686–706.

Aziz, K., and Settari, A. (1979). *Petroleum reservoir simulation*. Elsevier Publishing Co., New York, N.Y.

Barrett, R., et al. (1993). *Templates for the solution of linear systems: Building blocks for iterative methods*. Society for Industrial and Applied Mathematics Publications, Philadelphia, Pa.

Chaudhry, M. F. (1993). *Open-channel flow*. Prentice-Hall, Inc., Englewood Cliffs, N.J.

Chow, V. T., and Ben-Zvi, A. (1973). "Hydrodynamic modeling of two-dimensional water flow." *J. Hydr. Div.*, ASCE, 99(11), 2023–2040.

Cordes, C., and Putti, M. (1996). "Triangular mixed finite elements versus the finite volumes in groundwater modeling." *Int. Conf. Comp. Meth. Water Res. XII*, Computational Mechanics, Inc., Southampton, London, 61–68.

Department of Defense groundwater modeling system (GMS). (1995). Computer Graphics Laboratory, Brigham Young University, Utah.

Dongarra, J. (1993). *Performance of various computers using standard linear equation software*. Computer Science Department, University of Tennessee, Knoxville, Tenn., CS-89-85.

Dongarra, J., Lumsdaine, A., Pozo, R., and Remington, K. (1995). "IML++ v. 1.1. Iterative methods library." *Nat. Inst. of Standards and Technol. Rep.*, http://math.nist.gov, University of Tennessee, Knoxville, Tenn.

Fennema, R. J., and Chaudhry, M. H. (1990). "Explicit methods for 2D transient free-surface flows." *J. Hydr. Engrg.*, ASCE, 116(8), 1013–1034.

Fennema, R. J., Neidrauer, C. J., Johnson, R. A., McVicor, T. K., and Perkins, W. A. (1994). "A computer model to simulate natural everglades hydrology." *Everglades, the ecosystem and its restoration*, S. M. Davis and J. C. Ogden, eds., St. Lucie Press, Delray Beach, Fla., 249–289.

Fenner, R. T. (1975). *Finite element method for engineers*. MacMillan, London, England.

Fread, D. L. (1973). "Effects of time step in implicit dynamic routing." *Water Resour. Bull.*, 9(2), 338–350.

Fread, D. L. (1988). *The NWS DAMBRK model*. National Weather Service, National Oceanic and Atmospheric Administration, Silver Spring, Md.

Garcia, R., and Kahawita, R. A. (1986). "Numerical solution of the St. Venant equations with the McCormack finite difference scheme." *Int. J. Numer. Methods in Fluids*, 6, 507–527.

Hirsch, C. (1988). *Numerical computation of internal and external flows*. John Wiley & Sons, Inc., New York, N.Y.

Hromadka II, T. V., and Lai, C. (1985). "Solving the two-dimensional diffusion flow equations." *Proc. of the Spec. Conf. Sponsored by the Hydr. Div. of the ASCE*, American Society of Civil Engineers, New York, N.Y., 555–561.

Hromadka II, T. V., McCuen, R. H., and Yen, C. C. (1987). "Comparison of overland flow hydrograph models." *J. Hydr. Res.*, ASCE, 113(11), 1422–1440.

Kadlec, R. H., and Knight, R. L. (1996). *Treatment wetlands*. Lewis Publishers, Boca Raton, Fla.

Katopodes, N. D., and Strelkoff, R. (1978). "Computing two-dimensional dam break flood waves." *J. Hydr. Div.*, ASCE, 104(9), 1269–1288.

Lal, A. M. W. (1998). "Performance comparison of overland flow algorithms." *J. Hydr. Engrg.*, ASCE, 124(4), 342–349.

Lal, A. M. W., and Belnap, M. (1997). *A user's guide to HSE, the hydrologic simulation engine of the south Florida regional simulation model*. South Florida Water Managment District, West Palm Beach, Fla.

Liggett, J. A., and Woolhiser, D. A. (1967). "Difference solutions of the shallow-water equation." *J. Engrg. Mech. Div.*, ASCE, 93(2), 39–71.

McDonald, M., and Harbaugh, A. (1984). *A modular three dimensional finite difference groundwater flow model*. U.S. Geological Survey, Reston, Va.

Panton, R. L. (1984). *Incompressible flow*. John & Wiley Sons, Inc., New York, N.Y.

Ponce, V. M., Li, R.-M., and Simons, D. B. (1978). "Applicability of kinematic and diffusion models." *J. Hydr. Engrg.*, ASCE, 104(3), 353–360.

Raviart, P. A., and Thomas, J. M. (1977). "A mixed finite element method for second order elliptic problems." *Mathematical aspects of the finite element method*, I. Galligani and E. Magenes, eds., Springer-Verlag, New York, 61–68.

Seager, M. L. (1988). *SLAP, Sparse linear algebra package 2.0*. Lawrence Livermore National Lab., Livermore Computing Center, January 1986 Tentacle.

Smith, B. F., McInne, L. C., and Gropp, W. D. (1995). *PETSc 2.0 user*

manual. Tech. Rep. No. ANL-95/11, Argonne National Lab., Argonne, Ill.

Strelkoff, T., Schamber, D., and Katopodes, N. (1997). "Comparative analysis of routing techniques for the flood wave from a ruptured dam." *Proc. of Dam-Break Flood-Routing-Model Workshop held in Bethesda, Md.*, Water Resources Council, U.S. Department of Commerce, National Technical Information Service, Springfield, Va., 227–291.

Tan, W. (1992). *Shallow water hydrodynamics*. Elsevier Publishing Co., New York, N.Y.

Wang, H. (1982). *Introduction to groundwater modeling*. W. H. Freeman Co., New York, N.Y.

Xanthopoulos, T., and Koutitas, C. (1976). "Numerical simulation of a two-dimensional flood wave propagation due to dam failure." *J. Hydr. Res.*, Delft, The Netherlands, 14(4), 321–331.

Zhao, D. H., Shen, H. W., Tabios III, G. Q., Lai, J. S., and Tan, W. Y. (1994). "Finite-volume two-dimensional unsteady-flow model for river basins." *J. Hydr. Engrg.*, ASCE, 120(7), 863–883.

APPENDIX II. NOTATION

The following symbols are used in this paper:

E = energy head (m);
$\bar{\mathbf{F}}_r$ = average flux vector across wall r;
$\hat{\mathbf{F}}_k$ = flux vector at node k;
g = gravitational acceleration;
$\mathbf{H}$ = average water levels of all cells, in vector form (m);
$\hat{H}$ = water levels at nodes (m);
h = depth of water (m);
K = hydraulic transmissivity (m^2/s);
$\mathbf{M}$ = matrix obtained after linearizing $\mathbf{Q}$;
$\mathbf{n}$ = unit normal to wall;
n_b = Manning roughness coefficient;
$\mathbf{Q}(\mathbf{H})$ = inflow into all cells, in vector form;
Q_s = flow rate across structure;
$\mathbf{S}$ = source or sink terms for all cells, in vector form;
$\mathbf{s}_f$ = friction slope vector;
$\mathbf{s}_n$ = slope of water surface or energy surface;
$\mathbf{V}$ = flow velocity vector;
u, v = x and y components of flow velocity (m/s);
x, y = space coordinates (m);
$\hat{x}$, $\hat{y}$ = nodal coordinates;
z = ground elevation above datum (m);
$\Delta\mathbf{A}$ = diagonal matrix with cell areas at diagonals;
ΔA_i = area of cell i;
$\Delta\hat{A}_i$ = area of shadow cell i;
Δd_{mn} = distance between circumcenters of triangles m and n;
Δl_r = length of wall r;
Δt = time step (s); and
δ_s = slope below which only an approximate Manning equation is used.