PERFORMANCE COMPARISON OF OVERLAND FLOW ALGORITHMS

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

ABSTRACT: Two regional models, the South Florida Water Management Model (SFWMM) and the Natural System Model (NSM), are applied to analyze and predict water conditions in the Everglades and South Florida. Both of these models use an alternating direction explicit (ADE) type method to solve diffusion flow. In this paper, three finite-difference algorithms based on explicit, alternating direction implicit (ADI) and successive over relaxation (SOR) methods are examined as possible replacements for the ADE method. Various model solutions are verified using an axisymmetric test problem that is solved using an axisymmetric test model. A comparison of run time versus error plots proved that the ADI method has the best overall performance. The study includes a description of the relationship between the accuracies and run times of different algorithms and their spatial and temporal discretizations in dimensionless form. Linear and spectral analyses are used to derive theoretical expressions for numerical error, run time, and stability. Comparisons indicate that theoretical estimates of numerical error and run times closely approximate the experimental values. Results of this study are valuable as methods to determine optimum space and time discretizations of future modeling applications when the maximum allowable numerical error and the dimensions of the flow features to be simulated are known. Results can also be used to understand the magnitudes of numerical errors in existing modeling applications.

INTRODUCTION

The overland flow component of hydrologic and hydraulic models is commonly simulated by solving the St. Venant equations or their approximate forms. Overland flow plays a significant part in the hydrologic process in the Everglades and other areas of South Florida. Selection of the proper discretization and algorithm for such models is important, to obtain sufficiently accurate results at the required resolution with a minimum run time.

A number of algorithms have been used in the past to solve the complete St. Venant equations that govern overland flow. Chow and Ben-Zvi (1973) and Katopodes and Strelkoff (1978) developed early solution methods based on the finite-difference method, whereas Fenner (1975) and others developed approaches that used the finite-element method. Higher-order methods, based on the MacCormack method (Garcia and Kahawita 1986) and the finite-volume method (Zhao et al. 1994) have recently been used to improve the accuracy of rapidly varying flow. When the dynamic component is significantly large, as occurring in rapidly rising water levels and dam-break flows, the complete dynamic equations must be solved. The time step required for most of these explicit models is governed by the Courant condition, which sometimes result in long run times.

Areas such as South Florida are characterized by large areal extent, low slopes, widespread ponding, and slow regional flow dynamics. Kinematic wave models are inadequate for these cases because they neglect backwater effects. Dynamic models are, however, not necessary according to the theoretical conditions laid out by Ponce et al. (1978). Diffusion flow models have been found to be capable of simulating these regional flow conditions accurately (Fennema et al. 1994). Akan and Yen (1981), Hromadka and Lai (1985), and others have shown that the diffusion flow equation can accurately represent many of the flow situations found in nature. Diffusion flow models essentially neglect the inertia terms in the St. Venant equations. The diffusion flow equation can be written in terms of discharge, for hydrologic applications, and in terms of flow depth for hydraulic applications.

In addition to the time-step limitation under explicit conditions, most dynamic models introduce numerical errors and instabilities when the cell size is small, and the topographic data show relatively large variations in depth over these short cell lengths (Tan 1992). Zhao et al. (1994) used a Riemann solver to handle this case as a discontinuity. Smoothing is an alternative method that can be used to prevent oscillations. Diffusion flow models do not have this problem because inertia terms are not considered in the solution. Diffusion flow models only have one partial differential equation to solve for water level, compared with three coupled partial differential equations required for complete 2D dynamic flow using St. Venant equations. Unless the dynamic nature of flow conditions dictate otherwise, diffusion flow models offer computational advantages for areas such as South Florida. The conditions under which complete St. Venant equations have to be used, instead of diffusion equations, can be determined using the guidelines proposed by Ponce et al. (1978).

A number of methods have been used to solve 2D diffusion flow equations that are written in terms of depth. Xanthopoulous and Koutitas (1976) used an explicit method. Hromadka and Lai (1985) used an explicit method with linearization as carried out by Akan and Yen (1981) in their 1D model. The Natural System Model (NSM) and the South Florida Water Management Model (SFWMM) developed by SFWMD (Fennema et al. 1994) use a modified version of the alternating direction explicit (ADE) method to solve 2D diffusion flow. Almost any numerical method that is used to solve linear diffusion equations can also solve the linearized form of the 2D diffusion flow equations. The methods available include relaxation methods with different acceleration algorithms, and the alternating direction implicit (ADI) method that uses the Thomas algorithm to solve the tridiagonal matrix (Press et al. 1989).

The demand is increasing to obtain more accurate and detailed flow solutions from regional models for South Florida. Selection of the proper spatial and temporal discretizations is important to apply available computer resources most effectively. One purpose of this study is to compare the performances of different 2D diffusion flow models based on numerical errors and run times. The study includes stability, error, and a run-time analysis using both theoretical and experimental methods. Results of the study are useful as a means to determine the optimum space and time discretizations for new models and to understand the numerical errors in existing models.

[1] Lead Engr., South Florida Water Mgmt. Dist., 3301 Gun Club Rd., West Palm Beach, FL 33416.

Note. Discussion open until September 1, 1998. 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 March 11, 1996. This paper is part of the ***Journal of Hydraulic Engineering***, Vol. 124, No. 4, April, 1998. ©ASCE, ISSN 0733-9429/98/0004-0342–0349/$4.00 + $.50 per page. Paper No. 12889.

J. Hydraul. Eng. 1998.124:342-349.

GOVERNING EQUATIONS

Overland flow can be simulated by applying the depth averaged flow equations that are commonly referred to as St. Venant equations. These equations consist of a continuity equation and two momentum equations. The 2D continuity equation for shallow water flow is

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

where u and v = velocities in the x and y directions, respectively; h = water depth; RF = rainfall; IN = infiltration; and ET = evapotranspiration. RF, IN, and IN are expressed in units of length/time. Using water depth as a variable, the momentum equations in x and y directions are

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

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

where S_{fx}, S_{fy} = friction slopes in x and y directions, respectively. After neglecting the first three terms that contribute to inertia effects, the momentum equation in x and y directions reduces to $\partial H/\partial x = -S_{fx}$ and $\partial H/\partial x = -S_{fy}$, in which $H = h + z$ = water level above a datum and z = bottom elevation above the datum. The Manning's equation $V = 1/nh^{2/3}S_f^{1/2}$ is written in the direction of flow, in which n = Manning's coefficient; $V = \sqrt{v^2 + u^2}$ = magnitude of the velocity vector and S_f = friction slope. In diffusion flow, $S_f = S_s$ = slope of the water surface computed as $\sqrt{(\partial H/\partial x)^2 + (\partial H/\partial y)^2}$.

Hromadka and Lai (1985) showed that the flow velocity components u and v can be expressed in the following form using the Manning's equation:

$$u = -\frac{h^{2/3}}{n\sqrt{S_s}}\frac{\partial H}{\partial x} = -\frac{K}{h}\frac{\partial H}{\partial x} \qquad (4)$$

$$v = -\frac{h^{2/3}}{n\sqrt{S_s}}\frac{\partial H}{\partial y} = -\frac{K}{h}\frac{\partial H}{\partial y} \qquad (5)$$

in which K is expressed as

$$K = \frac{h^{5/3}}{n\sqrt{S_s}} \quad \text{for} \quad |S_s| > \delta \quad \text{and} \quad h > h_{\min} \qquad (6)$$

$$K = 0 \quad \text{otherwise} \qquad (7)$$

If a generic equation of the type $V = 1/nh^{\gamma}S_s^{\lambda}$ is used instead of the Manning's equation, K in (6) becomes $1/nh^{\gamma+1}S_s^{\lambda-1}$. Many wetland flows require an equation in this form. The condition $h \leq h_{\min}$ is used to facilitate wetting and drying, and δ is used to maintain K within finite limits. $\delta \approx 1.0 \times 10^{-7}$ and $h_{\min} = 0$ are used in the study. K is useful in linearizing and simplifying the diffusion flow equations. Without the source terms, the continuity (1) for the governing equation can be expressed using (4) and (5) as

$$\frac{\partial H}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial H}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial H}{\partial y}\right) \qquad (8)$$

K is a measure of the hydraulic conductivity in ground water equations. Source terms are not considered in the current study. The linearized form of the overland flow equation is similar to the ground water flow equation. Any of a wide choice of algorithms can be used to solve the linearized equation.

NUMERICAL ALGORITHMS

Four different finite-difference methods for solving the 2D overland flow equations are evaluated. Because the governing equation is based on conservation principles, a finite-volume-type formulation was used in this study. With rectangular grids the formulation is similar to the MAC (marker and cell) type formulation that was used by Xanthopolous and Koutitas (1976) and uses the average water stage of cell (i, j), denoted by $H^n_{i,j}$, to compute flows across cell boundaries. In the derivation, the eastern cell boundary of cell (i, j), for example, is marked as (i + 1/2). The continuity equation for cell (i, j) gives the following expression:

$$H^{n+1}_{i,j} = H^n_{i,j} + \alpha Q_{\text{net}}(H^{n+1})\frac{\Delta t}{\Delta A} + (1 - \alpha)Q_{\text{net}}(H^n)\frac{\Delta t}{\Delta A} \qquad (9)$$

in which $\Delta A = \Delta x \times \Delta y$ = area of the cell; Q_{net} = net inflow to the cell as a function of heads; α = weighting factor for semi-implicit schemes; α = 0, 1, and 0.5 for explicit, implicit, and Crank Nicholson type schemes, respectively; and n = time step. Q_{net} is computed as

$$Q_{\text{net}} = K_{i+1/2,j}(H_{i+1,j} - H_{i,j}) + K_{i-1/2,j}(H_{i-1,j} - H_{i,j})$$
$$+ K_{i,j+1/2}(H_{i,j+1} - H_{i,j}) + K_{i,j-1/2}(H_{i,j-1} - H_{i,j}) \qquad (10)$$

in which K values are obtained using (6) and (7). Explicit methods and linearized implicit methods use K values of the previous time step. When computing K at the east face (i + 1/2, j), for example, $h_{i+1/2,j} = 0.5(h_{i,j} + h_{i+1,j})$. An evaluation of S_s at cell faces requires at least three grid points. In the case of the east face (i + 1/2, j) of cell (i, j), for example,

$$S_s^2(\text{east}) = \frac{1}{\Delta x^2}\left[(H_{i+1,j} - H_{i,j})^2 + \frac{r^2}{16}(H_{i+1,j+1} + H_{i,j+1} - H_{i+1,j-1} - H_{i,j-1})^2\right] \qquad (11)$$

in which $r = \Delta x/\Delta y$. The ADE method uses the following expressions for S_s for both east and south faces:

$$S_s^2 = \frac{1}{\Delta x^2}[(H_{i+1,j} - H_{i,j})^2 + r^2(H_{i,j} - H_{i,j-1})^2] \qquad (12)$$

Boundary conditions have to be defined at all outside faces of boundary cells. If $H_{1,j}$ is an example of a boundary cell (1, j) with a cell wall facing west, a known head boundary condition is expressed as $H_{1,j} = f(\text{time})$. A known discharge at the face is expressed using (9) and (10) for mass balance, replacing $K_{1/2,j}(H_{0,j} - H_{1,j})$ with the known discharge Q_B. A no-flow boundary condition is derived by assigning $Q_B = 0$. With ADI methods, the same no-flow boundary can be approximated as $H^{n+1}_{1,j} = H^{n+1}_{2,j}$.

Explicit Method

The explicit method uses (9) and (10) with $\alpha = 0$ to compute $H^{n+1}_{i,j}$. The entire right-hand side is known from the previous time step including the K values. The time step is limited by the stability condition.

ADE Method

The overland flow algorithm used in the NSM and SFWMM models for South Florida uses a modified ADE method (Fennema et al. 1994). It has added features that allow the use of time steps exceeding the stability limit for explicit schemes. Stability is achieved by limiting the volume of water that enters or leaves a cell to the volume available in the cell that would not create a reverse water surface gradient. The following computations are carried out within a time step, in the sequence shown, using updated values at each step:

$$\begin{cases} H^*_{i,j} = H_{i,j} + \dfrac{V_{i+1/2,j}}{\Delta A} \\ H^*_{i+1,j} = H_{i+1,j} - \dfrac{V_{i+1/2,j}}{\Delta A} \\ H^{**}_{i,j} = H^*_{i,j} + \dfrac{V_{i,j-1/2}}{\Delta A} \\ H^{**}_{i,j-1} = H^*_{i,j-1} - \dfrac{V_{i,j-1/2}}{\Delta A} \end{cases}$$

$$\text{for} \quad j = 2, 3, \ldots M, \quad i = 1, 2, \ldots N - 1 \tag{13}$$

in which $V_{i+1/2,j}$ and $V_{i,j-1/2}$, etc., = volumes of water crossing from cells in the east and the south into cell (i, j); N, M = number of cells in x and y directions, respectively; and ΔA = area of a cell = $\Delta x \Delta y$. The superscript * shows the updated values at any given step.

$$V_{i+1/2,j} = \text{sign}(H^*_{i+1,j} - H^*_{i,j})\min[0.5\Delta A(|H^*_{i+1,j} - H^*_{i,j}|),$$
$$K_{i+1/2,j}\Delta t(|H^*_{i+1,j} - H^*_{i,j}|), h^*_{i+1,j}\Delta A] \tag{14}$$

$$V_{i,j-1/2} = \text{sign}(H^*_{i,j-1} - H^*_{i,j})\min[0.5\Delta A(|H^*_{i,j-1} - H^*_{i,j}|),$$
$$K_{i,j-1/2}\Delta t(|H^*_{i,j-1} - H^*_{i,j}|), h^*_{i,j-1}\Delta A] \tag{15}$$

in which the first term inside square parentheses gives the volume needed to prevent reverse flow gradients. The second term gives the volume computed using the Manning's equation. The third term provides the volume available in the source cell. This algorithm is unconditionally stable even at large time steps. However, there is a propagation error that occurs with large time steps, because a disturbance can travel only one cell length during one time step. The numerical errors are studied in the error analysis that follows.

ADI Method

The ADI method is the most efficient among all of the solution methods considered. The mass balance condition in (9) and (10) with $\alpha = 0.5$ forms the basis for the ADI formulation. The differencing operators used in the derivation are as follows:

$$D_x H^n_{i,j} = 0.5 \frac{\Delta t}{\Delta A} [K_{i+1/2,j}(H^n_{i+1,j} - H^n_{i,j}) + K_{i-1/2,j}(H^n_{i-1,j} - H^n_{i,j})] \tag{16}$$

$$D_y H^n_{i,j} = 0.5 \frac{\Delta t}{\Delta A} [K_{i,j+1/2}(H^n_{i,j+1} - H^n_{i,j}) + K_{i,j-1/2}(H^n_{i,j+1} - H^n_{i,j})] \tag{17}$$

Eqs. (9) and (10) can now be expressed using the operators as

$$(1 - D_x - D_y)H^{n+1}_{i,j} = (1 + D_x + D_y)H^n_{i,j} \tag{18}$$

By neglecting higher-order terms, (18) can be solved by solving the following split formulations in sequence:

$$(1 - D_x)H^*_{i,j} = (1 + D_y)H^n_{i,j} \tag{19}$$

$$(1 - D_y)H^{n+1}_{i,j} = (1 + D_x)H^*_{i,j} \tag{20}$$

Eqs. (19) and (20) are solved as two 1D problems for each row and column of the 2D domain using the Thomas algorithm for tridiagonal matrices. For example, the lower diagonal, diagonal, and upper diagonal elements for (19) can be expressed as

$$A_{i,j} = -0.5 \frac{\Delta t}{\Delta x \Delta y} K_{i-1/2,j} \tag{21}$$

$$B_{i,j} = 1 + 0.5 \frac{\Delta t}{\Delta x \Delta y} (K_{i+1/2,j} + K_{i-1/2,j}) \tag{22}$$

$$C_{i,j} = -0.5 \frac{\Delta t}{\Delta x \Delta y} K_{i+1/2,j} \tag{23}$$

Right-hand sides of (19) and (20) consist entirely of known values at time step n.

Linear Successive Over Relaxation Method

The linear successive over relaxation (SOR) method is based on the weighted implicit formulation shown in (9) and (10). It is shown later that large time steps can be used when $\alpha > 0.5$. The accuracy can be improved by using $\alpha = 0.5$ as in the Crank-Nicholson-type methods. Eqs. (9) and (10) are used to derive the following equation used in the SOR algorithm:

$$a_{i,j}H^{n+1}_{i+1,j} + b_{i,j}H^{n+1}_{i-1,j} + c_{i,j}H^{n+1}_{i,j+1} + d_{i,j}H^{n+1}_{i,j-1} + e_{i,j}H^{n+1}_{i,j} = f_{i,j} \tag{24}$$

in which

$$a_{i,j} = \frac{\alpha \Delta t}{\Delta A} K_{i+1/2,j} \tag{25}$$

$$b_{i,j} = \frac{\alpha \Delta t}{\Delta A} K_{i-1/2,j} \tag{26}$$

$$c_{i,j} = \frac{\alpha \Delta t}{\Delta A} K_{i,j+1/2} \tag{27}$$

$$d_{i,j} = \frac{\alpha \Delta t}{\Delta A} K_{i,j-1/2} \tag{28}$$

$$e_{i,j} = -(a_{i,j} + b_{i,j} + c_{i,j} + d_{i,j}) - 1 \tag{29}$$

$a_{i,j}$, $b_{i,j}$, . . . , $f_{i,j}$ are determined at time n. The SOR method using Chebyshev acceleration and odd-even ordering, as explained by Press et al. (1989), is used to solve the system of equations in (24). With odd-even ordering, one half-sweep is carried out at odd points, and the other half-sweep is carried out updating even points. As part of the acceleration process, an optimal over relaxation is determined during the computer run. A convergence criterion based on the magnitude of the correction is used to terminate the relaxations.

A nonlinear SOR method very similar to the linear SOR method also can also be used to obtain the solution. With such a method, $a_{i,j}$, $b_{i,j}$, . . . are updated with every iteration along with K. The nonlinear SOR method is relatively inefficient. The assumption of K as a constant within a time step is found to be valid for many test problems.

STABILITY AND ERROR ANALYSIS

Stability of diffusive overland flow algorithms is studied using the Von Neuman method. The error analysis is carried out using spectral analysis. In the error analysis, the solution of the equation is assumed as a "sine" function in complex variable form. The behavior of the scheme in response to the function is compared with the behavior of the governing equation in response to the same function. Assuming a solution domain of length L_x in the x direction, a mesh spacing of Δx only allows a minimum wavelength $\lambda_{\min} = 2\Delta x$ and a maximum wave length of $2L_x$. An arbitrary ith harmonic in the solution over the grid can be represented by a wave number, $k_{xi} = i\pi/L_x$ with wavelength $\lambda_{xi} = 2L_x/i$. Wave number k is defined as $2\pi/\lambda$. In the analysis, a term ϕ_x representing the ith harmonic is defined as (Hirsch 1989)

$$\phi_x = k_{xi}\Delta x = \frac{i\pi}{N} \tag{30}$$

in which ϕ_x = the phase angle. $\phi_x = 0$ and $\phi_x = \pi$ correspond to the lowest and highest frequencies in the solution in the x direction. π/ϕ_x gives an estimate of the level of discretization measured as the number of discretizations per half sine wave of the solution. ϕ_x is the nondimensional form of Δx, and a ϕ_y can be defined similarly in the y direction.

Using ϕ_x and ϕ_y, a single (i, j)th harmonic at time step n is represented by $E^n_{i,j}e^{Ii\phi_x}e^{Ij\phi_y}$, in which $E^n_{i,j}$ = amplitude; and $I = \sqrt{-1}$. Assuming that the diffusion equation is linearized as in (8) and discretized with (9) and (10), the following expression can be derived for the amplification factor ρ when $\Delta x = \Delta y$ and $\phi_x = \phi_y = \phi$ (Hirsch 1989):

$$\rho = \frac{E^{n+1}}{E^n} = \frac{1 - 4d(1 - \alpha)\beta \sin^2(\phi/2)}{1 + 4d\alpha\beta \sin^2(\phi/2)} \tag{31}$$

in which $\beta = K\Delta t/\Delta x^2$ = nondimensional form of Δt; $d = 1, 2$ for 1D and 2D problems. The stability of the scheme based on linear analysis is determined by $|\rho| \le 1$ or $\beta \le 1/2d(1 - 2\alpha)$. This condition can be related to a time-step restriction for explicit problems as

$$\Delta t \le \frac{1}{2}\frac{\Delta x^2}{K}\frac{1}{1 + r^2} \tag{32}$$

in which $r = \Delta x/\Delta y$. Preceding equations show that weighted schemes are unconditionally stable for $\alpha > 0.5$.

The amplitude E^{n+1} after one time step of length Δt of the numerical algorithm can be compared with the exact solution for the same duration for one Fourier component of the initial solution. A component that satisfies the governing equation is

$$H(x, t) = H_0 \exp[-K(k_x^2 + k_y^2)t]e^{Ik_x x}e^{Ik_y y} \tag{33}$$

where H_0 is obtained from Fourier decomposition of the initial condition. The same initial condition when subjected to the numerical scheme given by (9) and (10) gives an error. The error in amplitude after one time step, expressed as a function of nondimensional Δt and Δx is (Hirsch 1989)

$$\varepsilon = 1 - \frac{1 - 4d(1 - \alpha)\beta \sin^2(\phi/2)}{1 + 4d\alpha\beta \sin^2(\phi/2)}\frac{1}{e^{-d\beta\phi^2}} \tag{34}$$

The error of fully implicit and explicit models is obtained by expanding (34)

$$\varepsilon = \pm\frac{d^2\beta^2\phi^4}{2} + \frac{d\beta\phi^4}{12} + \ldots = \pm\frac{d^2k^4K^2\Delta t^2}{2} + \frac{dKk^4\Delta t\Delta x^2}{12} + \ldots \tag{35}$$

where + and − signs correspond to implicit and explicit models, respectively. For Crank-Nicholson type schemes, the error in the 2D solutions is

$$\varepsilon = \frac{\beta\phi^4}{6} - \frac{\beta\phi^6}{180} + \ldots \tag{36}$$

For 1D models, this error becomes $\varepsilon = \beta\phi^4/12 - \beta\phi^6/36 + \cdots$. Because the error is determined by ϕ and β, in which $\beta = (h^{5/3}/n\sqrt{S_s})/(\Delta t/\Delta x^2)$, the model error can be easily related to its parameters and discretizations. If Manning's n is doubled, for example, Δt can also be doubled without affecting the error. If h is doubled, Δt has to be reduced to 31% of its value.

The numerical error in the final 2D solution ε_T after a simulation time T depends on the number of time steps n, and the error at each time step ε. ε_T is bound by $n_t\varepsilon$, in which $n_t = T/\Delta t$, and is called the maximum error in the study

$$\varepsilon_T \approx \frac{\varepsilon}{\beta\phi^2}TKk^2 \tag{37}$$

where k = wave number of the solution. In the case of fully implicit and explicit methods

$$\varepsilon_T \approx 2TKk^2\phi^2(\pm\beta + 1/12) \tag{38}$$

in which + and − signs correspond to explicit and implicit methods, respectively. Eq. (38) shows that the maximum error in the final solution is approximately proportional to β starting with an offset. The error increases with T, K, k, and ϕ.

Run Times of Algorithms

Run times and error norms are used to determine performances of different algorithms. Computer time for a model is proportional to the number of cells and time steps. For a hypothetical case with N and M grid cells in the X and Y directions and n_t time steps, simulation time $T = n_t\Delta t$; the spatial domain dimensions are $L_x = N\Delta x$ and $L_y = M\Delta y$. Total CPU time t_r for the explicit method is c_1n_tNM or $(c_1)(T/\Delta t)(L_x/\Delta x)(L_y/\Delta y)$, in which c_1 = a computer dependent constant representing run time per cell per time step. Assuming $\Delta x = \Delta y$, and a given constant numerical error target

$$t_r = c_1\frac{T}{\beta}\frac{(NM)^2K}{L_xL_y} = \frac{c_1TKAk^4}{\beta\phi^4} \tag{39}$$

in which A = area simulated by the model; T = period of simulation; and k = wave number of an arbitrary Fourier component in the solution. Eq. (39) shows that the run time increases rapidly as ϕ decreases or the level of discretization increases.

Analytical error estimates in the study are compared with numerical error estimates for a test problem. The numerical estimates are based on the largest error defined as $\|\varepsilon\|_\infty = \max|\varepsilon_i|$, $1 \le i \le N$, where ε_i = error at grid point $i = 1, \ldots N$ of the solution.

NUMERICAL EXPERIMENTS

Numerical experiments are used to verify the accuracy of various numerical models developed in the study and to verify the accuracy of the theoretical estimates of error and run time. An axisymmetric test problem is selected for this purpose because it can be solved with both the 2D and the axisymmetric models. The latter is developed by slightly modifying a 1D model similar to the model by Akan and Yen (1981). The 2D models using ADI and SOR methods were implemented with $\alpha = 0.5$ for higher accuracy.

A test problem representing conditions of a sinusoidal water surface profile and a flat bottom is used in the study because the error analysis is based on sinusoidal functions. The experiment is designed to mimic some of the flow features observed in parts of South Florida. The test problem involves flow over a 160.93 × 160.93 km² (100 × 100 mi²) area. The initial condition at $t = 0$ is

$$H = \left[0.4575 + 0.1525\cos\left(\frac{\pi r}{r_{\max}}\right)\right] m \quad \text{for} \quad r \le r_{\max} \tag{40}$$

$$H = 0.305m \quad \text{otherwise} \tag{41}$$

in which r = distance from the domain center; $r_{\max}$ = 32,187.9 m. A boundary stage of 0.305 m is maintained and the simulation period T is 12 d. The boundary is far enough away from where the flow occurs that the effect on the solution is assumed to be negligible. The Manning's coefficient used is 1.0. RF, I, and ET are neglected in the test. The axisymmetric model is based on the axisymmetric continuity equation for shallow water flow with no source and sink terms given by

$$\frac{\partial(hr)}{\partial t} + \frac{\partial(uhr)}{\partial r} = 0 \tag{42}$$

where r = radial distance. The modification of the 1D diffusion flow model to axisymmetric conditions is accomplished simply by multiplying the linearized $K_{i+1/2}$ by $r_{i+1/2}/r_i$. The axisymmetric problem is solved using resolutions Δr = 32.187 m, Δt = 25.92 s, which are much higher than the resolutions possible with the 2D models under the same conditions. The 2D model requires extremely large run times with these resolutions. With the 2D model, ϕ = 0.31 is used, which corresponds to Δx = 3,218.7 m and N = 50. The numerical solution with the axisymmetric model is assumed to be much closer to the exact solution than the 2D solution because the axisymmetric solution remains within 1×10^{-7} m with higher resolutions. Because the axisymmetric model has also been verified previously under 1D conditions, numerical error analysis of the 2D model is carried out assuming that the errors in the axisymmetric model are very small compared with the errors to be analyzed.

RESULTS AND DISCUSSION

The test results with the 2D model matched closely with each other and with the axisymmetric solution. Table 1 shows selected results from the final solution obtained with less than 4 h of run time using a SUN Sparc 20. The water level shown is at the center of the domain after 12 d. Fig. 1(a) shows a contour plot of the stages obtained using a 50 × 50 grid and the 2D explicit model. The figure shows a circular distribution of flow. Fig. 1(b) shows the stages at a cross section through the center. The plots obtained for all of the methods show negligible differences from each other.

The numerical error analysis is carried out by running the models with different combinations of times and space discretizations. The computer run time needed for each of the models is measured using a SUN Sparc workstation 20 (speed 90 MHz, 4.1 Mflops/s measured with linpack benchmark test, Dongarra 1993) running in single precision. Fig. 2 shows the variation of the run time with model error of the final solution ε_T. The computed model errors in the figure are presented by expressing $\|\varepsilon\|_\infty$ as a fraction of the maximum depth at t = 0, which is 0.61 m. All of the plots in Fig. 2 are obtained for a constant ϕ = 0.31 or a constant 50 × 50 cell configuration over the area. The curves show, in general, that more computer time is needed to obtain higher accuracies. Fig. 2 shows that the ADI method is the most accurate for a given run time. Next to ADI, explicit and ADE methods perform well at high- and low-accuracy ranges respectively. The linearized SOR method falls in the midrange of all methods with regard to combined high accuracy and fast run time. The curved nature of an SOR method shows the effect of the adaptive relaxation exhibited by the method. The relatively larger scatter in the measurement of small run times and errors also have to be taken into account in interpreting results in these ranges. Fig. 3 shows the variation of the curve in Fig. 2 for the explicit model when ϕ or Δx is varied. Note that the run time varies inversely with $\beta\phi^4$ as explained by (39). In Fig. 3, very small ϕ values contribute more to excessive run times than to improve the accuracy.

Solid symbols in Fig. 4 show the approximately linear variation of the error in the final solution ε_T with β when ϕ = 0.31. The scatter shows the effect of dynamic instabilities on the error estimates. Because local K or β are not constant in diffusion flow models, approximate values are used in the study, computed using the largest water depth and slope at t = 0. These estimates are adequate for the order-of-magnitude-type error analysis. Local K or β values can be extremely large

TABLE 1. Selected Results of Simulated Water Levels at Center of Test Problem after 12 d

Method (1)	Level at center (m) (2)	Δt (s) (3)	Δx (m) (4)
Axi-sym	0.442105	25.92	32.19
Explicit	0.442171	25.31	804.67
ADE	0.44103	506.25	1,609.35
ADI	0.44222	8,100.0	804.67
SOR	0.44264	2,025.0	3,218.7

Note: Δx and Δt are selected to limit the run time in the Sparc 20 to less than 4 h.

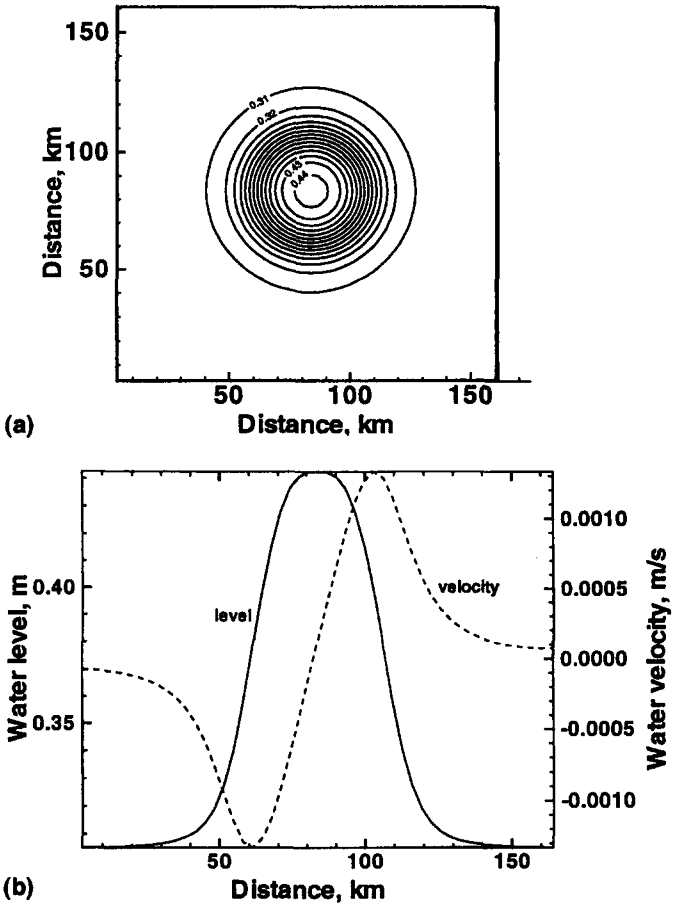

FIG. 1. (a) Contour Plot of Water Surface Elevations after 12 d; (b) Cross Section of Fig. 1(a) through Center after 12 d

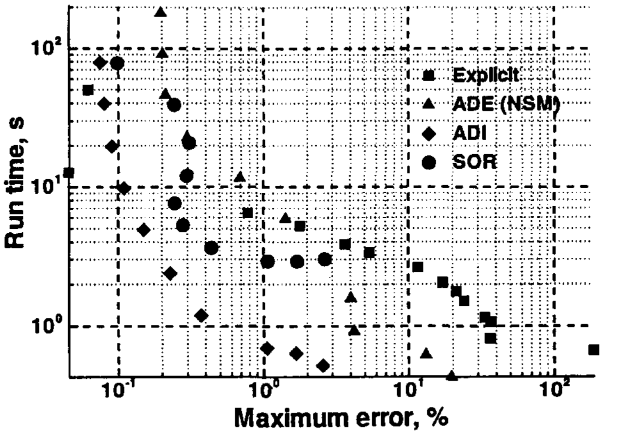

FIG. 2. Run Times Needed for Test Problem at Different Error Levels

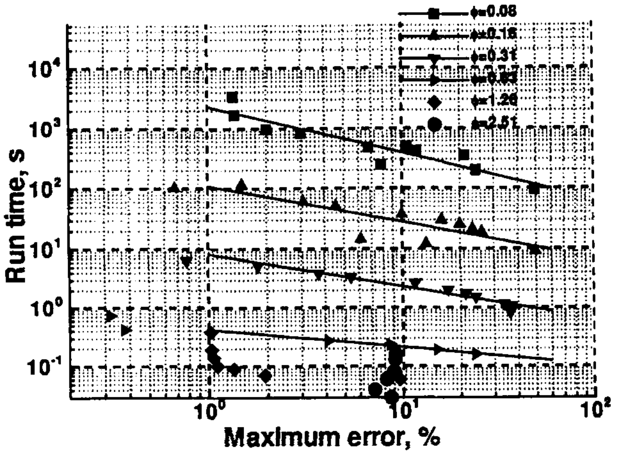

FIG. 3. Variation of Run Time with Maximum Error and φ for Explicit Model

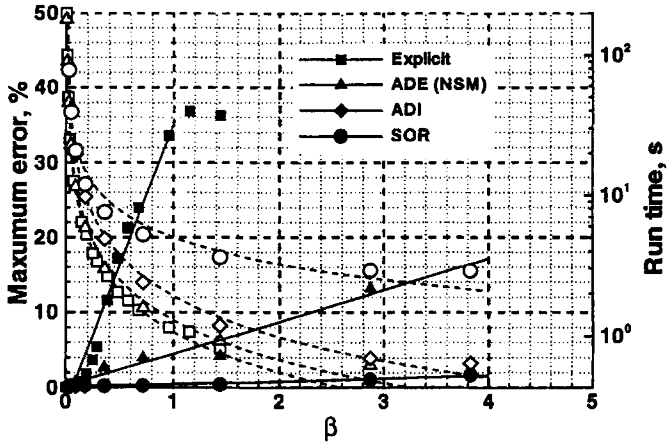

FIG. 4. Variation of Maximum Errors (Solid Symbols) and Run Times (Empty Symbols) with β Using All Four Models when φ = 0.31. Solid and Dashed Lines Show Trends in Errors and Run Times, Respectively

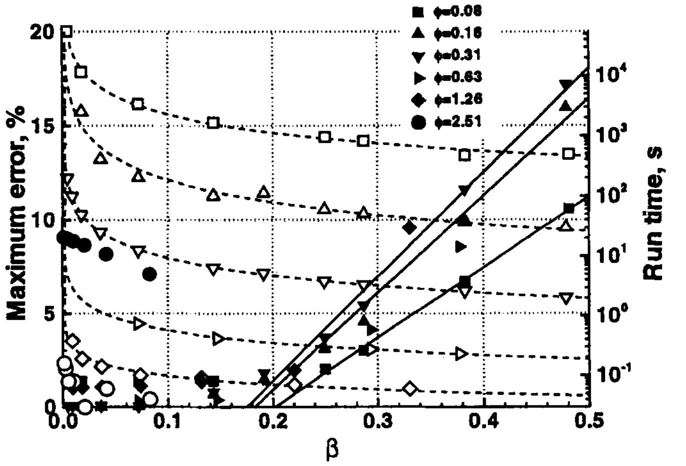

FIG. 5. Variation of Maximum Errors (Solid Symbols) and Run Times (Empty Symbols) of Explicit Method with β and φ. Solid and Dashed Lines Show Trends in Errors and Run Times Respectively Determined Using Regression

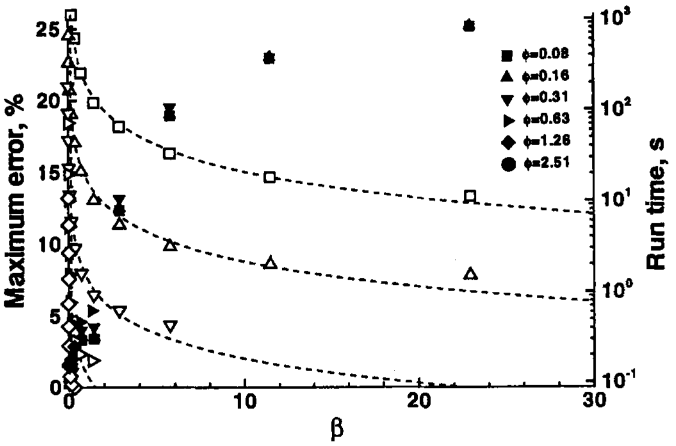

FIG. 6. Variation of Maximum Errors (Solid Symbols) and Run Times (Empty Symbols) of ADE Method with β and φ. Dashed Lines Show Trends in Run Times

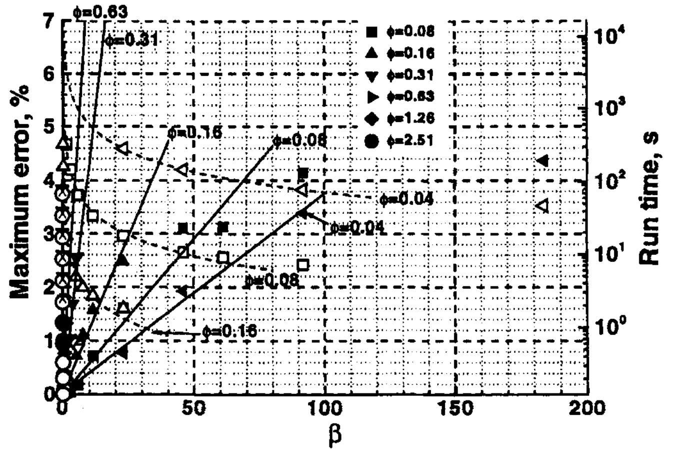

FIG. 7. Variation of Maximum Errors (Solid Symbols) and Run Times (Empty Symbols) of ADI Method with β and φ. Solid and Dashed Lines Show Trends in Errors and Run Times Respectively Obtained Using Regression

at near-zero slopes and they cause local instabilities even with very small Δt. However, these local instabilities stay contained unless Δt is large. Fig. 4 shows that errors of ADI and SOR methods are low and equal over a large range of β because both are based on weighted implicit formulations with α = 0.5. When using ADI and SOR methods, β values as large as 200 give small errors. Note that the ADI and SOR symbols coincide. With the explicit model, instability begins at the crest and outer fringe of the sinusoidal solution when $\beta \approx 0.1$. With the ADI method, signs of tiny wiggles show up at the fringe at $\beta \approx 2$. These instabilities grow extremely slowly with increasing β and remain localized for the most part even when β values are as large as 200.

Fig. 4 also shows that run times of the models when ϕ = 0.31. For a given β, Fig. 4 shows that the run time of the ADI model is only slightly higher than the run time of the ADE and explicit models. Eqs. (38) and (39) show the analytical relationships for error and run time. Both analytical relationships and results from model runs show approximately linear variations of the error with β starting with an offset. Figs. 5–8 for explicit, ADE, ADI, and SOR models are similar to Fig. 4, except that they are plotted for a number of ϕ values. The lines in these figures illustrate the trends in the data points.

The analytical expressions for numerical errors and run times given by (37) and (39) are compared to the actual errors and run times of the numerical models in Fig. 9. Only the ADI model comparison is shown in Fig. 9. Using regression analysis, (39) can be fit to the explicit and SOR models using c_1 = 3.4×10^{-5} and 6.1×10^{-5}, respectively, in the case of a Sparc 20. Fig. 9 shows that even if β is computed using rough values of model parameters, the analytical expressions are capable of explaining the behavior and the order of magnitude of model errors and run times.

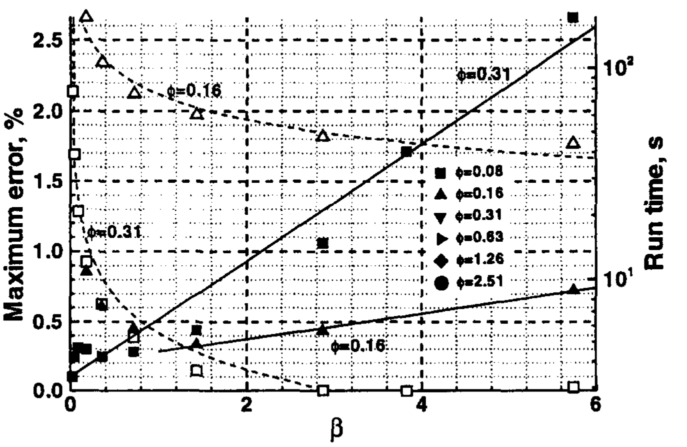

FIG. 8. Variation of Maximum Errors (Solid Symbols) and Run Times (Empty Symbols) of SOR Method with β and ϕ. Solid and Dashed Lines Show Trends in Errors and Run Times Respectively Obtained Using Regression

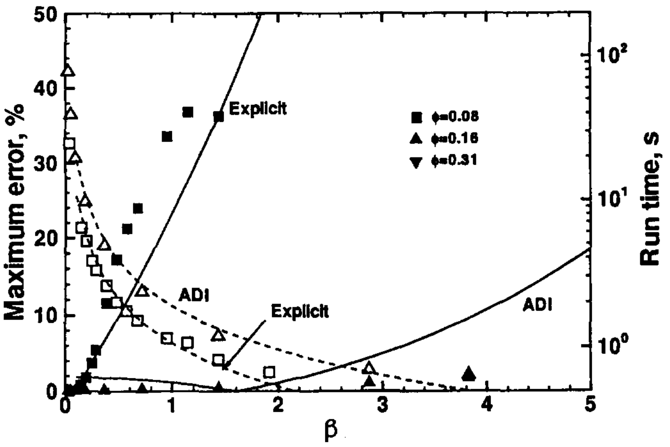

FIG. 9. Analytical and Experimental Estimates of Numerical Errors and Run Times. Solid and Empty Symbols Show Model Errors and Run Times Obtained by Running Model. Solid and Dashed Lines Show Analytical Estimates of Model Errors and Run Times, Respectively

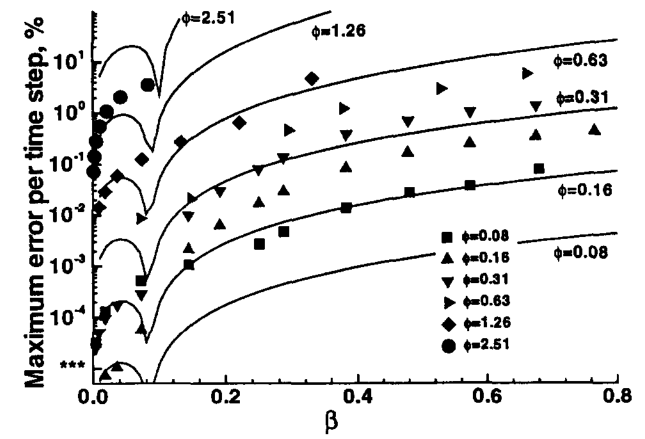

FIG. 10. Analytical and Experimental Estimates of Numerical Errors (Solid Lines and Symbols, Respectively) in Explicit Solution, for One Time Step

Errors presented in Figs. 2–9 are measured at the end of the model run and, therefore, depend on the number of time steps. Figs. 10 and 11 show the errors derived for one time step, for explicit, and ADI models. These results are independent of run times and, therefore, can be applied to any model using similar algorithms. Errors per time step are approximately computed by dividing the errors at the end of the simulation by the number of time steps. Figs. 10 and 11 show that the theoretical expression in (34) is capable of explaining the behavior of the error and its order of magnitude.

The highest accuracy of a wave component in a solution obtained using a given spatial discretization ϕ has a limit. Figs. 4–7 show that even with extremely small time steps, an error remains in the solution with large ϕ, and that it cannot be reduced unless ϕ itself is reduced. With ϕ = 2.67, for example, the solution error is more than 8% even when the smallest possible β value is used. With ϕ = 0.65, it is possible to reduce the error to about 2%. Fig. 12 shows the maximum error in the solution at different resolutions when using explicit and ADI methods. In the figure, resolution is also expressed as the number of discretizations for a half-sine-wave, computed as π/ϕ. Fig. 12 is useful in deciding the coarsest spatial resolution of a model possible when it is required to simulate a flow feature of a known wave number with a known accuracy.

The results of the study are useful in selecting a solution algorithm and in obtaining optimal discretizations for a diffusion flow model. First using Fig. 12 an upper limit of Δx or ϕ is determined that can limit the maximum error in the highest frequency component of the solution to the required design value. β is selected next for a given method using Figs. 5–8 or (37). If the run time computed for the method using (39) is excessive, β or ϕ, or both, have to be adjusted to obtain a balance between minimizing the error and the run time. If the run time is only slightly excessive, it is sufficient to change β

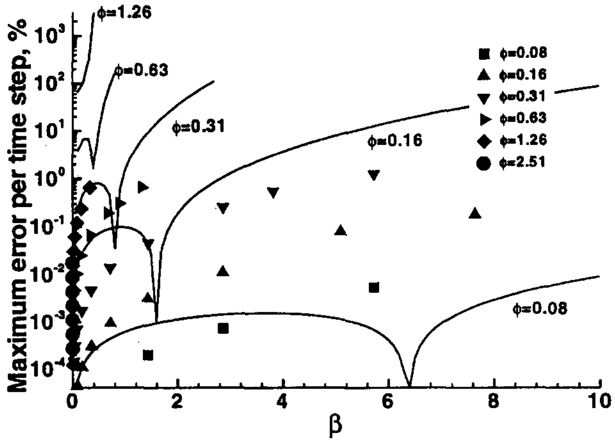

FIG. 11. Analytical and Experimental Estimates of Numerical Errors (Solid Lines and Symbols, Respectively) in ADI Solution, for One Time Step

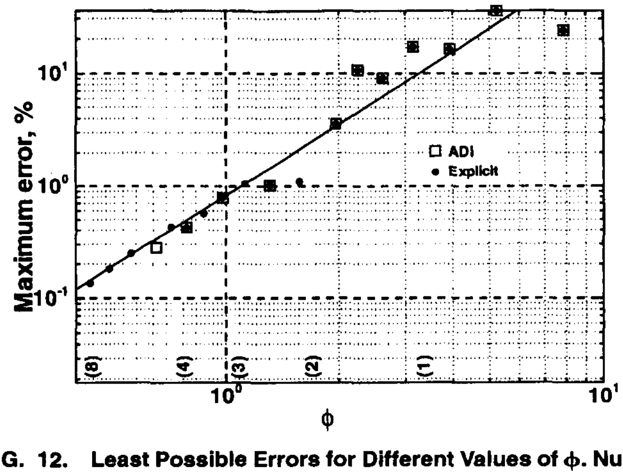

FIG. 12. Least Possible Errors for Different Values of ϕ. Number of Discretizations per Half Sine Wave (π/ϕ) Are Shown in Parentheses

alone. If run times are extremely high, ϕ has to be increased to achieve a smaller run time, and Fig. 12 gives the smallest wave that can be simulated by the model with the new ϕ with an acceptable accuracy. ϕ and β are finally converted to dimensional forms Δx and Δt using physical characteristics of the model domain. Results of the study are also useful in analyzing errors in existing models. After computing nondimensional ϕ and β for an existing model, the order of magnitude of the error can be determined using the results of the study.

CONCLUSIONS

A number of numerical models that use the explicit, ADI, and the SOR methods are developed to solve 2D diffusion flow equations. The accuracies of these models are verified using an axisymmetric flow problem and an axisymmetric diffusion flow model that has been verified for 1D problems. A plot of run time versus model error is used to compare model performances within various error ranges. The plots show that the ADI method is more efficient than the other algorithms considered. The ADE method has a relatively short run time and may be appropriate when high accuracy is not a priority. Explicit methods require long run times, but they are more accurate. SOR methods can be efficient under certain conditions because of the use of the adaptive relaxation parameter.

Analytical expressions were derived in this study to describe amplification errors and run times, in terms of nondimensional space and time discretizations. These expressions were shown to accurately describe the behavior of these same properties in the numerical models. These expressions are capable of explaining the behavior and the order of magnitude of model errors and run times. The results also provide maximum discretizations (ϕ or Δx) that can still maintain given levels of accuracy in the high-frequency components of the solution. Results of the study are useful in selecting the optimal discretizations for new overland flow models or estimating the approximate numerical errors of existing models. Results of the study confirm that numerical errors in diffusion flow models increase with decreasing bed roughness, surface slope, and wavelength of the water surface profile, and increasing depth.

ACKNOWLEDGMENTS

The writer wishes to thank Joel VanArman, Todd Tisdale, Jayantha Obeysekera Brion Lehar, Randy Van Zee, Ray Santee, and Bob Laura of the South Florida Water Management District, and Marshall Taylor of the Resources Planning Association for reviewing the paper and making valuable comments.

APPENDIX I. REFERENCES

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

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

Documentation of the South Florida Water Management Model. (1997). Hydrologic Systems Modeling Division, Planning Dept., South Florida Water Management Dist., West Palm Beach, Fla., 1997.

Dongarra, J. J. (1993). "Performance of various computers using standard linear equation software." *CS-89-85*, Computer Sci. Dept., Univ. of Tennessee, Knoxville, Tenn.

Fennema, R. J., Neidrauer, C. J., Johnson, R. A., MacVicor, 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, Fla., 249–289.

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

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

Hirsch, C. (1989). *Numerical computation of internal and external flows.* John Wiley & Sons, Inc., New York, N.Y.

Hromadka, T. V., II, and Lai, C. (1985). "Solving the two-dimensional diffusion flow model." *Proc., Spec. Conf.*, sponsored by the Hydr. Div. of ASCE, Lake Buena Vista, Fla.

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

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

Press, W. P., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1989). "Numerical recipes." *The art and science of computing*, Cambridge University Press, Cambridge, U.K.

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

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

Zhao, D. H., Shen, H. W., Tabios, G. Q., III, 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:

A = area of model domain;
D_x, D_y = finite-difference operators in x and y direction, (16);
$E^n_{i,j}$ = amplitude of wave related to cell (i, j) at time step n;
ET = evapotranspiration;
g = gravitational acceleration;
H = water level above datum;
$H^n_{i,j}$ = average water level of cell (i, j) or row i column j at time step n;
h = water depth;
I = $\sqrt{-1}$;
IN = infiltration;
K = constant, $h^{5/3}/(n\sqrt{S_s})$ when Manning's equation is used and when finite;
$K_{i+1/2,j}$ = transmissivity K between cells (i, j) and (i + 1, j);
k = wave number, $2\pi/\lambda$;
L_x, L_y = length of spatial domain in x and y directions;
N, M = number of discretizations in x and y directions;
n = time step;
Q_{net} = net inflow into cell;
RF = rainfall intensity;
r = radial distance;
S_{fx}, S_{fy} = friction slopes in x and y directions;
S_s = maximum slope of water surface;
T = total run time;
t = time;
u, v = water velocities in x and y directions;
V = magnitude of flow velocity vector;
$V_{i+1/2,j}$ = volume of water transferred between cells (i, j) and (i + 1/2, j);
x, y = Cartesian coordinates;
z = ground elevation above datum;
α = time-weighting factor;
β = nondimensional time step = $K\Delta t/\Delta x^2$ when $\Delta x = \Delta y$;
ΔA = area of cell;
Δt = time step;
Δx, Δy = length of spatial discretization;
δ = slope below which flow rate can be neglected in area;
ε = percentage error in one time step;
λ = wavelength of Fourier component of solution; and
ϕ_x, ϕ_y = dimensionless space discretization or phase angle for x and y discretizations, $k_{xi}\Delta x$ and $k_{yi}\Delta y$.