

# A 2D implicit time-marching algorithm for shallow water models based on the generalized wave continuity equation

Kendra M. Dresback<sup>\*,†</sup>, Randall L. Kolar and J. Casey Dietrich

*School of Civil Engineering and Environmental Science, University of Oklahoma, Norman, OK 73019, U.S.A.*

## SUMMARY

This paper builds upon earlier work that developed and evaluated a 1D predictor–corrector time-marching algorithm for wave equation models and extends it to 2D. Typically, the generalized wave continuity equation (GWCE) utilizes a three time-level semi-implicit scheme centred at  $k$ , and the momentum equation uses a two time-level scheme centred at  $k + \frac{1}{2}$ . It has been shown that in highly non-linear applications, the algorithm becomes unstable at even moderate Courant numbers. This work implements and analyses an implicit treatment of the non-linear terms through the use of an iterative time-marching algorithm in the two-dimensional framework. Stability results show at least an eight-fold increase in the maximum time step, depending on the domain. Studies also examined the sensitivity of the  $G$  parameter (a numerical weighting parameter in the GWCE) with results showing the greatest increase in stability occurs when  $1 \leq G/\tau_{\max} \leq 10$ , a range that coincides with the recommended range to minimize errors. Convergence studies indicate an increase in temporal accuracy from first order to second order, while overall error is less than the original algorithm, even at higher time steps. Finally, a parallel implementation of the new algorithm shows that it scales well. Copyright © 2004 John Wiley & Sons, Ltd.

**KEY WORDS:** shallow water equations; finite elements; generalized wave continuity equation; implicit time-marching; parallel computing

## 1. INTRODUCTION

In a previous article [1], we discussed and analysed a predictor–corrector time-marching algorithm (abbreviated predictor–corrector algorithm or just PC, herein) in a one-dimensional (1D) setting utilizing the finite element framework and the generalized wave continuity equation (GWCE). This paper builds upon that work and extends it to a two-dimensional (2D) setting, wherein it is assessed for stability, accuracy, parameter sensitivity, and parallel efficiency.

\*Correspondence to: K. M. Dresback, School of Civil Engineering and Environmental Science, University of Oklahoma, Norman, OK 73019, U.S.A.

†E-mail: dresback@ou.edu

Contract/grant sponsor: National Science Foundation; contract/grant numbers: ACI-9623592 and EEC-9912319  
Contract/grant sponsor: United States Department of Education

Herein, the finite-element code is based on Lynch and Gray's [2] wave continuity equation (WCE), which suppresses spurious oscillations without having to dampen the solution either numerically or artificially. Further studies by Kinnmark [3] determined that there was no loss in the wave propagation characteristics of the WCE if a numerical parameter,  $G$ , is introduced. This parameter expresses a balance between the primitive form and the pure wave form of the shallow water equations. The model utilized in this paper, ADCIRC (an ADvanced three-dimensional CIRculation model) [4] is based on the GWCE.

Currently, non-linear applications with ADCIRC have stability problems unless a severe Courant number restriction is imposed. The Courant number is defined as

$$C_r = \frac{c \times \Delta t}{\Delta x} \quad (1)$$

where  $c = \sqrt{gh}$  is the linear wave celerity,  $\Delta x$  is minimum node spacing and  $\Delta t$  is the time step. In practice, we have found that for deep ocean flows, a practical upper bound of the Courant number ( $C_r$ ) is 0.5 in order to maintain stability; however, an even tighter constraint (e.g.  $C_r \ll 0.1$ ) must be imposed if the simulation includes barrier islands, constricted inlets, or wetting and drying of near-shore elements. In order to relax this restriction, an alternative time-marching procedure was proposed that treats the non-linear terms implicitly [5].

As reported in Reference [1] (and repeated here for completeness), a number of earlier studies looked at time-marching in shallow water models, but often from a noise suppression point-of-view. For example, Lee and Froehlich [6] summarize several time-marching procedures in their shallow water equation review paper, which covers everything from the trapezoidal rule to three-level semi-implicit schemes. Lynch and Gray [7] showed several of the same time-marching procedures in greater detail. They indicate that the best scheme for finite element shallow water models is the three-level semi-implicit scheme. Several years later, Kinnmark and Gray [8] examined a semi-implicit wave equation that produced accurate results, yet still treated the non-linear terms explicitly. Most of the more recent work with GWCE-based models has focused either on incorporating more physics or minimizing spatial error, e.g. alternative meshing criteria [9–11], wetting and drying [12], treatment of boundary conditions [13, 14], 3D baroclinic simulations [15, 16], and more accurate estimates of the vertical velocity [17]. Furthermore, attempts to achieve timely simulations have led to parallel codes [18, 19]. Little recent work with GWCE-based models has been devoted to alternative time-marching algorithms. The intent of this study is to fill this gap, viz, an implicit treatment of non-linear terms in both the GWCE and the momentum equation.

An implicit treatment can be realized by either simultaneous integration of the full non-linear equations or a predictor–corrector algorithm. A predictor–corrector algorithm was chosen over the simultaneous integration for the following reasons: (1) it can be easily implemented within the framework of the existing ADCIRC code; (2) it minimizes the size of the matrices that must be stored and inverted; and (3) it is more computationally efficient than the simultaneous integration of the full non-linear equations.

In this paper, we examine the impact of the 2D predictor–corrector algorithm on stability,  $G$  sensitivity, and temporal accuracy, both globally and locally. Also, we implement a combined parallel/predictor–corrector algorithm and assess the scalability of the resulting code. In our earlier paper, we provided the background on the shallow water equations and indicated the proposed changes to the time-marching algorithm [1]; a summary of these sections is included herein for completeness.

2. SHALLOW WATER EQUATIONS

The full shallow water equations can be found in various sources [2–4, 20–22]; the GWCE and non-conservative form of the momentum (NCM) equation, which form the basis of the ADCIRC model, are given below. Using operator notation, where  $L$  represents the primitive continuity equation, and  $\mathbf{M}^C$  the conservative form of the momentum equation, we present the GWCE for depth-averaged flows as

$$W^G = \frac{\partial L}{\partial t} + GL - \nabla \cdot \mathbf{M}^C \tag{2}$$

where  $G$  is the numerical ‘penalty’ parameter. Lynch and Gray’s [2] WCE can be obtained by setting  $G = \tau$ , where  $\tau$  is the bottom friction. It should be noted that the higher the value of  $G$ , the more the GWCE approaches the primitive equation. Expanded versions of the GWCE and NCM equation are shown below. All terms are described in the nomenclature section, but the predominant variables are noted after the equations. The abbreviations appearing above certain terms in these equations will be discussed in subsequent sections.

*GWCE*

$$W^G = \frac{\partial^2 \zeta}{\partial t^2} + G \frac{\partial \zeta}{\partial t} + G \nabla \cdot \overset{\text{‘Gg’}}{(H\mathbf{v})} - \nabla \cdot \left\{ \overset{\text{‘ag and at’}}{\nabla \cdot (H\mathbf{v}\mathbf{v})} + \overset{\text{‘cg’}}{H\mathbf{f}} \times \mathbf{v} + \overset{\text{‘bg’}}{\tau H\mathbf{v}} \right. \\ \left. + H \nabla \left[ \frac{p_a}{\rho} + \overset{\text{‘fg’}}{g(\zeta - \alpha\eta)} \right] - \mathbf{A} - \frac{1}{\rho} \nabla \cdot (HT) \right\} = 0 \tag{3}$$

*NCM Equation*

$$\mathbf{M} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \overset{\text{‘am’}}{\nabla} \mathbf{v} + \overset{\text{‘bm’}}{\tau \mathbf{v}} + \mathbf{f} \times \mathbf{v} + \nabla \left[ \frac{p_a}{\rho} + g(\zeta - \alpha\eta) \right] - \frac{\mathbf{A}}{H} - \frac{1}{\rho H} \nabla \cdot (HT) = 0 \tag{4}$$

where  $\zeta$  is the elevation of the water surface above the datum,  $t$  is time,  $\mathbf{v}$  is the depth-averaged velocity, and  $H$  is the total fluid depth,  $h + \zeta$ .

Algorithms based on these two equations result in solutions that compare well with analytical solutions and field data for both elevation and velocity. The codes typically use equal-order finite element interpolating functions (linear  $C^0$  elements). As presently coded, semi-implicit time discretization of the GWCE uses a three-time-level approximation centred at  $k$ , while time discretization of the NCM equation uses a lumped two-time-level approximation centred at  $k + \frac{1}{2}$ . Equations are linearized by formulating the non-linear terms explicitly. Exact quadrature rules are used. Product terms in the equations are simplified by linearly interpolating the products of the variables, not the individual variables.  $L_2$  interpolation is applied to the advective terms. A time-splitting solution procedure is adopted wherein the GWCE is first solved for nodal elevations and then the NCM equation is solved for the velocity field. Resulting discrete equations can be found in Luettich *et al.* [4].

### 3. DESCRIPTION OF THE IMPLICIT TIME-MARCHING ALGORITHM

As noted, the current semi-implicit algorithm evaluates the linear terms implicitly and the non-linear terms explicitly. At the past and present time levels in ADCIRC, elevation and velocity values are known (either from initial conditions or previous calculations). The original algorithm takes the elevation and velocity values for the past ( $k - 1$ ) and the present ( $k$ ) and uses them to calculate the values for the future ( $k + 1$ ) time level for the linear terms. However, the non-linear terms are evaluated using only the elevation and velocity values at the present time level ( $k$ ). Kolar *et al.* [5] hypothesized that the stability constraint stems primarily from this explicit evaluation of non-linear terms.

In order to evaluate the non-linear terms implicitly, a predictor–corrector time-marching algorithm is introduced. The predictor stage, which is equivalent to the original algorithm, evaluates the non-linear terms using values from the present. Predicted future values, called  $k^*$ , and the already-known present ( $k$ ) and past ( $k - 1$ ) values are then used to obtain corrected values for the future ( $k + 1$ ) time level. The corrector stage can be repeated as many times as necessary until convergence. In all applications to date, a single iteration of the corrector stage appears to be sufficient.

Non-linear terms exist in both governing equations for ADCIRC—the NCM and GWCE. Our study focuses on all eight non-linear terms, identified in Equations (3) and (4) by the names above the terms. Six reside in the GWCE: advective (abbreviated ‘ag and at’), in which ‘ag’ is associated with the spatial derivative and ‘at’ is associated with the temporal derivative;‡ finite amplitude (abbreviated ‘fg’); Coriolis (abbreviated ‘cg’); GWCE flux times  $G$  (abbreviated ‘Gg’); and GWCE flux times  $\tau$  (abbreviated ‘bg’). Two are from the NCM equation: the advective term (abbreviated ‘am’) and the bottom friction term (abbreviated ‘bm’).

Through the use of time weight coefficients, users have the option to distribute the relative contribution of the non-linear terms over the three time levels. Comprehensive 1D studies [1] and accompanying 2D studies have shown that optimal coefficients are problem dependent, but that near-optimal results for any domain are found by centring the GWCE time weights at  $k$  (meaning that the time weights for the non-linear terms are weighted equally between  $k + 1$  (or  $k^*$ ),  $k$ ,  $k - 1$ ) and centring NCM time weights at  $k + \frac{1}{2}$  (meaning that the terms are weighted equally between  $k$  and  $k + 1$  (or  $k^*$ )). For this manuscript, all non-linear terms are treated implicitly using this near-optimal time weighting scheme.

### 4. TWO-DIMENSIONAL DOMAINS

We examine algorithm behaviour on a number of domains: the quarter annular harbour (denoted ‘quarter annular’), a fictional grid that has a well-documented analytical solution, and several application domains—Bight of Abaco (denoted ‘Bahamas’), Western North Atlantic (denoted ‘Eastcoast’), and Gulf of Mexico.

‡Earlier modifications to the ADCIRC code converted the advection term in the GWCE to non-conservative form by using the primitive continuity equation to split it into two terms, one involving a spatial derivative and one involving a temporal derivative. See Reference [22] for full details.

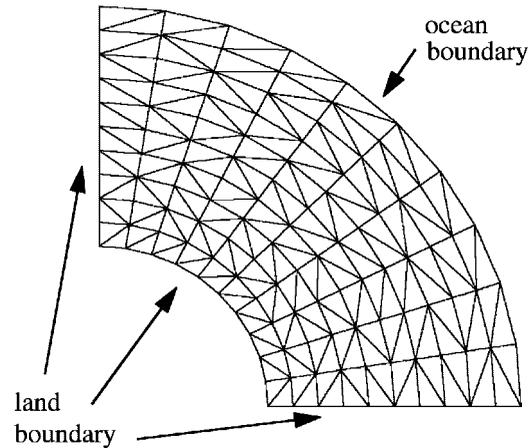


Figure 1. Quarter annular harbor domain ( $10 \times 10$  resolution).

The quarter annular harbour grid is shown in Figure 1. Boundaries are marked on the figure, with either ocean or land indicated. The boundary condition for the open ocean boundary is the  $M_2$  tidal constituent with a 1 m amplitude, while the land boundaries are no flow. Resolutions used in comparison studies include  $5 \times 5$ ,  $10 \times 10$ ,  $15 \times 15$ ,  $20 \times 20$ ,  $25 \times 25$  and  $30 \times 30$  (radial divisions  $\times$   $\theta$  divisions).

The application domains (Eastcoast, Gulf of Mexico and Bahamas) are shown in Figure 2; boundary conditions are indicated in the figure. Table I contains information regarding parameters, boundary, and grid information for each of the application domains. In all of the domains, the land boundaries are treated as no flow. Two meshing criteria were used in developing the Gulf of Mexico grids:  $\lambda/\Delta x$ , which is commonly used in grid development, and the Local Truncation Error Analysis (LTEA), developed by Hagen *et al.* [11]. In the LTEA technique, nodes are placed in order to minimize spatial truncation error.

## 5. NUMERICAL EXPERIMENTS AND DISCUSSION

### 5.1. Stability

Dominant non-linear terms preclude the use of traditional stability studies, such as Fourier analysis. Therefore, we utilize heuristic methods. In particular, stability changes with the new algorithm were determined from these steps: (1) Each domain was evaluated using the original algorithm to obtain the maximum stable time step (to nearest 5 s) for each type of spatial discretization; (2) Each domain was evaluated using the predictor–corrector algorithm to obtain the maximum stable time step for each spatial discretization; (3) Results from the two previous steps were compared to one another and a percent change between the two resulting time steps was obtained. Because the corrector iteration requires another solution of the system matrix, we need to achieve more than a  $n \times 100\%$  change for the predictor–corrector algorithm to be considered cost-effective, where  $n$  is the number of corrector steps. This is a conservative estimate because it assumes the entire load vector is reevaluated with

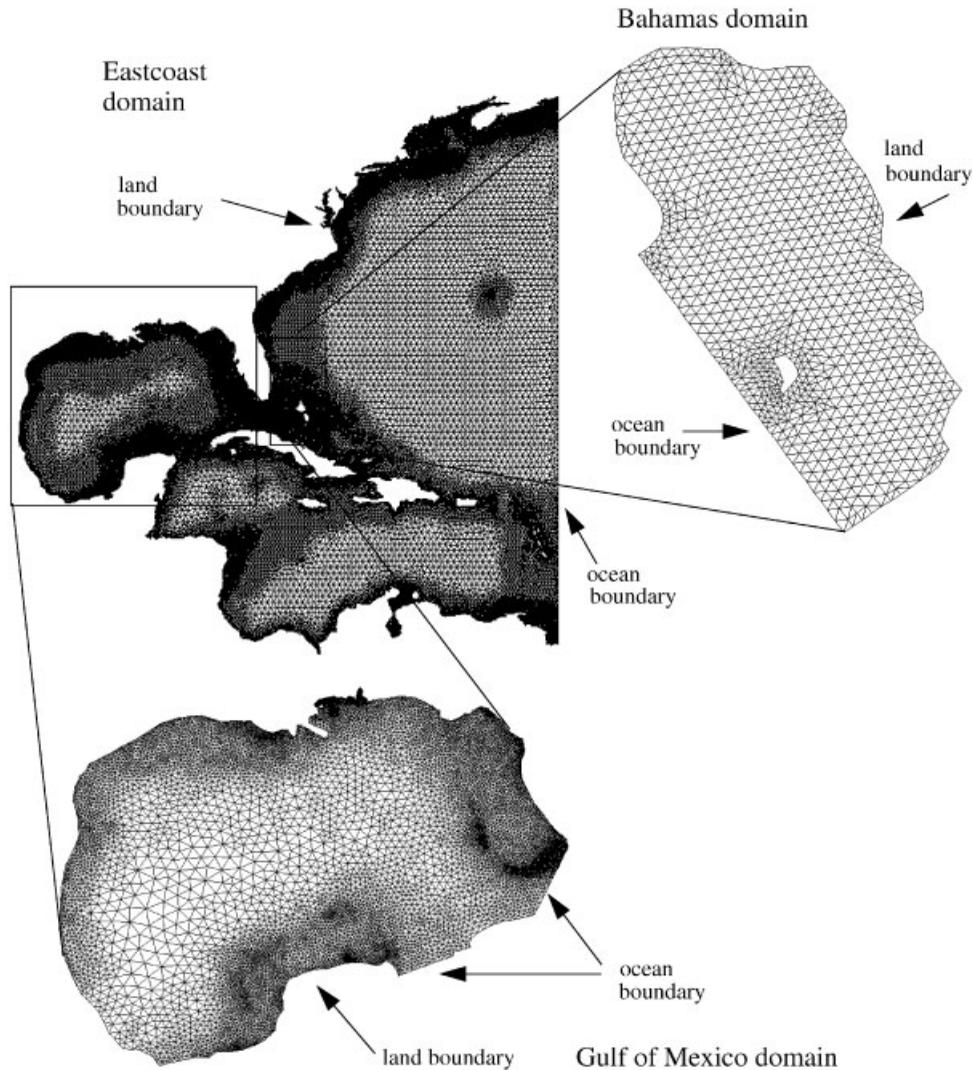


Figure 2. Eastcoast domain with the Gulf of Mexico (LTEA resolution) and Bahamas domains zoomed in.

each iteration, while in reality, only the  $k^*/k + 1$  portion of the non-linear terms needs to be updated. Previous results showed no significant gains in performing more than one iteration of the corrector step, therefore only one iteration is considered herein.

Results for each of the domains are summarized in Table II with the maximum allowable time steps shown along with the maximum Courant number. In all the domains, results show that the Courant number restriction relaxes with the predictor–corrector algorithm. Generally, Courant numbers with the original algorithm are less than 0.5, while Courant numbers

Table I. Application domain information.

	Bahamas	Gulf of Mexico		Eastcoast
Meshing criteria	$\lambda/\Delta x$	$\lambda/\Delta x$	LTEA	$\lambda/\Delta x$
# nodes	926	11701	11934	32947
# elements	1696	21970	22870	61705
Min. bathymetry (m)	1.0	1.0	0.7	3.0
Max. bathymetry (m)	~9.0	~3600.0	~3600.0	~6000.0
original $G$ value ( $s^{-1}$ )	0.009	0.009	0.009	0.005
$M_2$ amplitude (m)	0.395	0.07–0.173	0.07–0.173	0.0652–0.5580
$O_1$ amplitude (m)	0.075			
$K_1$ amplitude (m)	0.095			
$S_2$ amplitude (m)	0.06			
$N_2$ amplitude (m)	0.10			

Table II. Numerical stability experiments—maximum stable time step and associated Courant number for various domains.

Domain	Original $\Delta t$ , s ( $C_r$ )	Predictor–corrector $\Delta t$ , s ( $C_r$ )	Percent increase (%)
<i>Quarter annular</i>			
$5 \times 5$	2205 (0.50)	7095 (1.50)	222
$10 \times 10$	1120 (0.47)	5140 (2.14)	359
$15 \times 15$	730 (0.45)	4250 (2.47)	482
$20 \times 20$	550 (0.45)	3685 (3.00)	569
$25 \times 25$	445 (0.45)	3400 (3.40)	664
$30 \times 30$	370 (0.45)	3060 (3.74)	727
<i>Bahamas</i>			
$\lambda/\Delta x$	245 (0.57)	885 (2.06)	261
<i>Gulf of Mexico</i>			
$\lambda/\Delta x$	200 (0.52)	580 (1.54)	190
LTEA	100 (0.17)	360 (0.62)	260
<i>Eastcoast</i>			
$\lambda/\Delta x$	55 (0.032)	470 (0.27)	754

greater than 1.0 can be realized with the predictor–corrector algorithm. All domains obtain the necessary 100% increase for the new algorithm to be cost-effective. The greatest increase in the maximum stable time step occurs with the Eastcoast domain, which shows an eight-fold increase.

For the different spatial discretizations, results from the quarter annular and Gulf of Mexico domains indicate that resolution changes do affect the stability results. In particular for the quarter series, the maximum Courant number with the original algorithm remains constant with increasing resolution; however, the maximum Courant number with the predictor–corrector algorithm increases with increasing resolution. For the Gulf of Mexico domain, we evaluated two different meshing techniques, the typical  $\lambda/\Delta x$  and the LTEA. The LTEA method adds refinement in the grid where truncation errors are high, which usually coincides with the continental shelf and shelf break. For these two refinements, we analysed the spatial variability

of the Courant number over the continental shelf and shelf break. From these studies, we found that the LTEA grid tends to even out the magnitude of the Courant number over the break, with the most restrictive Courant number occurring in the shallow, near-shore region; in contrast, the  $\lambda/\Delta x$  grid shows a limiting Courant number at the shelf break.

### 5.2. *G Sensitivity*

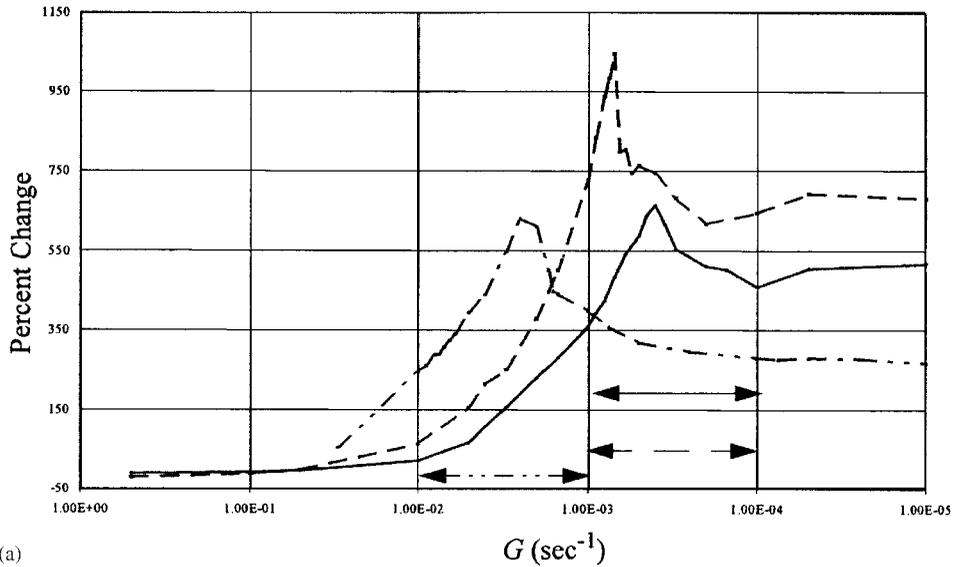
Sensitivity studies provide information on how parameter changes impact algorithm performance. Herein, we are especially interested in how  $G$ , the numerical parameter in the GWCE, impacts the maximum stable time step for the PC algorithm. Sensitivity analyses were conducted on three domains, with several spatial resolutions: quarter annular domain with two spatial resolutions,  $10 \times 10$  and  $30 \times 30$ ; Bahamas domain; and Gulf of Mexico domain with two meshing techniques,  $\lambda/\Delta x$  and LTEA. In each of these domains, we analysed a range of  $G$  values between  $0.00001$  and  $0.1 \text{ s}^{-1}$ . Using the PC algorithm, for each  $G$  parameter value we obtained the maximum stable time step, which we compared to the maximum stable time step from the original algorithm with the  $G$  parameter fixed at its base value.

Percent changes between these two scenarios are shown in Figure 3. Figure 3(a) shows the quarter annular domain with the two resolutions, along with Bahamas domain. For the quarter annular domains, the greatest increase in stability occurs with  $G$  between  $0.001$  to  $0.0001 \text{ s}^{-1}$ , with peaks at  $G = 0.0004 \text{ s}^{-1}$  for the  $10 \times 10$  resolution and  $G = 0.0007 \text{ s}^{-1}$  for the  $30 \times 30$  resolution. For the Bahamas domain, the greatest increase occurs between  $0.01$  and  $0.001 \text{ s}^{-1}$ , with the peak at  $0.0025 \text{ s}^{-1}$ . Figure 3(b) shows the results from the two Gulf of Mexico meshes. For the  $\lambda/\Delta x$  grid, the greatest increase in stability occurs when the  $G$  value is between  $0.001$  and  $0.0001 \text{ s}^{-1}$  with the peak occurring at  $G = 0.00068 \text{ s}^{-1}$ , while for the LTEA grid, the greatest increase in stability occurs between  $0.01$  and  $0.0001 \text{ s}^{-1}$  with two peaks—one at a  $G = 0.002 \text{ s}^{-1}$  and the other at  $G = 0.00045 \text{ s}^{-1}$ . A common feature of all is that stability is highly sensitive to the value of  $G$  used in the simulation.

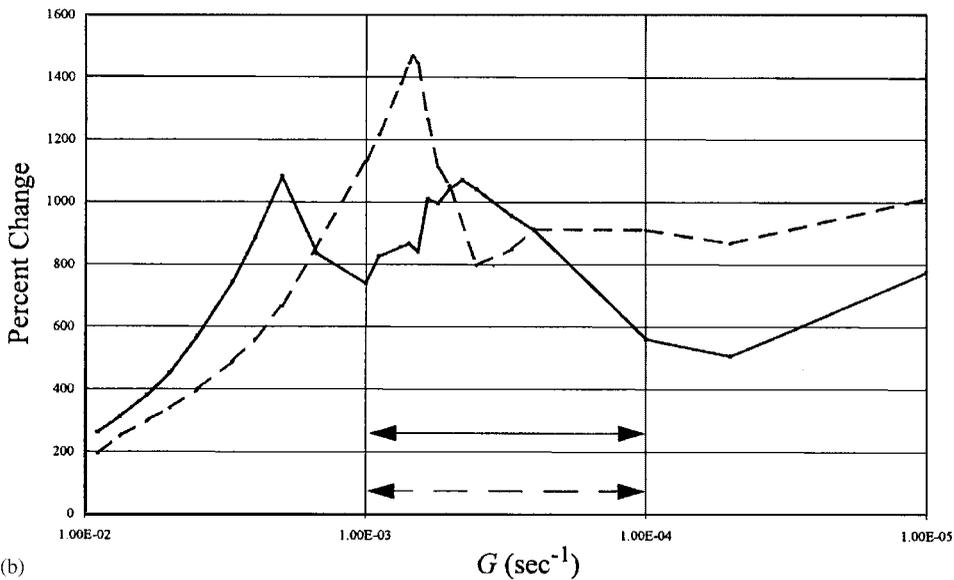
An optimum range for  $G/\tau_{\max}$  should lie between 1 and 10 in order to minimize the mass balance errors and errors in the generation of non-linear constituents, as indicated in previous work by Kolar *et al.* [22]. They also indicated that when  $G/\tau_{\max}$  increases above 10, oscillations can appear in the solution. We computed this ratio for each of the domains to determine if the stability peak lies in-between this recommended range. In Figure 3, we indicate with arrows where  $1 \leq G/\tau_{\max} \leq 10$  for each domain. In short, for all domains tested, the maximum increase in stability coincides with the range of  $G$  values that minimizes the mass balance errors and errors in the generation of non-linear constituents. Thus, a single value of  $G$  can meet both criteria.

### 5.3. *Temporal accuracy*

In this section, we investigate the influence of the predictor–corrector algorithm on temporal accuracy, both globally and locally. Globally, we analyse the temporal accuracy using  $L_2$  and  $L_\infty$  norms for both elevation and velocity fields and determine the changes to the order of accuracy between the two algorithms. Discussion of these results occurs in the first subsection. In the second subsection, cumulative area fraction errors (CAFE) plots [23] are used to study the local behaviour of temporal accuracy.



(a)



(b)

Figure 3. Sensitivity of the  $G$  parameter to stability for two-dimensional bathymetry: (a) quarter annular domains ( $10 \times 10$  (solid line) and  $30 \times 30$  (dashed line) resolution) and Bahamas domain (dot-dash line) and (b) Gulf of Mexico  $\lambda/\Delta x$  (dashed line) and LTEA (solid line). Arrows indicate where  $1 \leq G/\tau_{\max} \leq 10$  for each domain shown in the figures above.

5.3.1. *Global temporal accuracy (convergence rates).* In order to evaluate the global behaviour of temporal accuracy, including the overall convergence rate, we compared solutions from a coarse temporal resolution to a ‘true’ (fine  $\Delta t$ ) solution of 10 s. Accuracy

changes were quantified by analysing the global error, as measured by the  $L_2$  norm and the  $L_\infty$  norm for both elevation and velocity fields. For all domains, norms were evaluated at 120 discrete times covering 10 complete tidal cycles of the  $M_2$  tide, which is the dominant tidal signal. For the  $L_2$  norm, we averaged the results over time, while for the  $L_\infty$  norm, we determined the maximum value over time. We performed the temporal accuracy experiments on all of the domains: quarter annular ( $30 \times 30$  resolution), Bahamas, Gulf of Mexico ( $\lambda/\Delta x$ ) and Eastcoast. As representative examples, Figure 4 shows the elevation  $L_2$  norms for three of these domains: Bahamas (4a), Gulf of Mexico (4b), and Eastcoast (4c).

In all domains, results show that the error for the predictor–corrector algorithm plots below the original algorithm (less absolute error). The order of temporal accuracy, as indicated by the slope of the lines in Figure 4, increases from approximately first order for the original algorithm to second order for the predictor–corrector algorithm (see Table III). Results using either the  $L_\infty$  norm or the velocity field are similar.

These results mimic what we found with the 1D experiments [1]. Noteworthy is that the predictor–corrector algorithm shows the greatest increase in accuracy for domains where the bathymetry gradients are significant over much of the domain, such as the Gulf of Mexico. In contrast, the least gains are seen with the Eastcoast domain, which may be due to the fact that the majority of nodes are located in the deep bathymetry, where the wave propagation is more nearly linear. In these instances, updating the non-linear terms does not provide as much improvement to the accuracy.

*5.3.2. Local temporal accuracy.* Next, we evaluated the behaviour of the temporal accuracy over the spatial domain utilizing CAFE plots [23]. CAFE plots provide both absolute and relative errors between two simulations of the same domain with the same spatial resolution but different temporal resolutions. Absolute errors are obtained from the differences between results for a coarse and a fine temporal resolution, while relative errors take the absolute errors and divide by the results of the fine temporal resolution [11]. A frequency graph is developed based on these errors; viz. for each error level, the cumulative area fraction is computed, which is defined as the ratio of area of the grid associated with a certain error or greater is compared to the total area of the grid. The velocity field is decomposed into its elliptical trajectory that is described by the following components: major-semi axis, phase lag, eccentricity, and major-semi axis direction difference [11]. Elevation, being a scalar field, can be represented by amplitude and phase.

The following steps were used to develop CAFE curves for this study: (1) harmonic data is recorded for the original algorithm over several tidal cycles for coarse and fine temporal resolutions; (2) harmonic data is recorded for the predictor–corrector algorithm over several tidal cycles for coarse and fine temporal resolutions; (3) absolute and relative errors are calculated for each algorithm; (4) cumulative errors for both algorithms are computed; and (5) results from the original and predictor–corrector algorithms are plotted against each other to determine the percent area exceeding a certain error criteria.

Figure 5 shows two sample CAFE plots. To read these plots, one must first realize that a perfect solution (no error) would plot as a vertical line located at  $x=0$ , indicated by the vertical solid line in Figure 5. Any deviations from this line represent errors in the simulation; the further the graph is from  $x=0$ , the more the error. For a given convergence criteria, one finds the associated percent of cumulative area in the domain that exceeds the given criteria. For example in Figure 5(a), one first selects a convergence criteria (in this example,  $\pm 0.02\text{cm}$ )

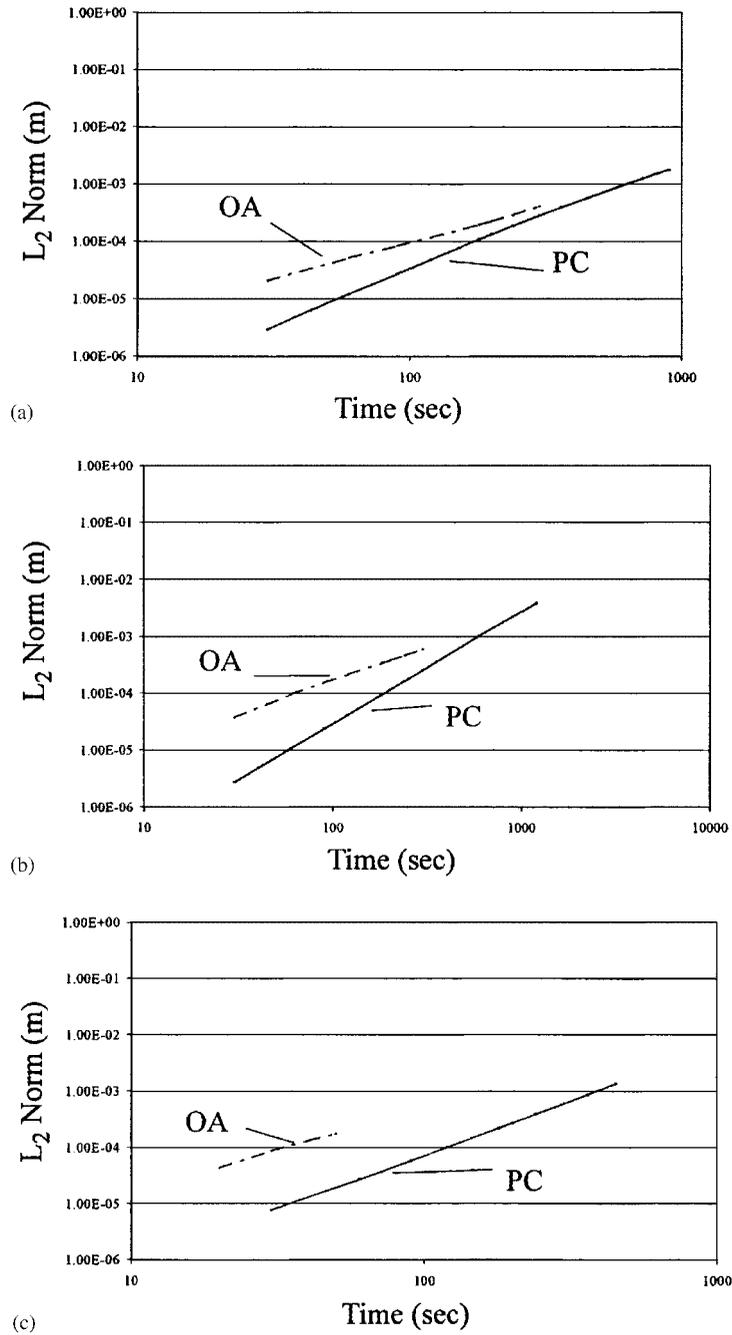
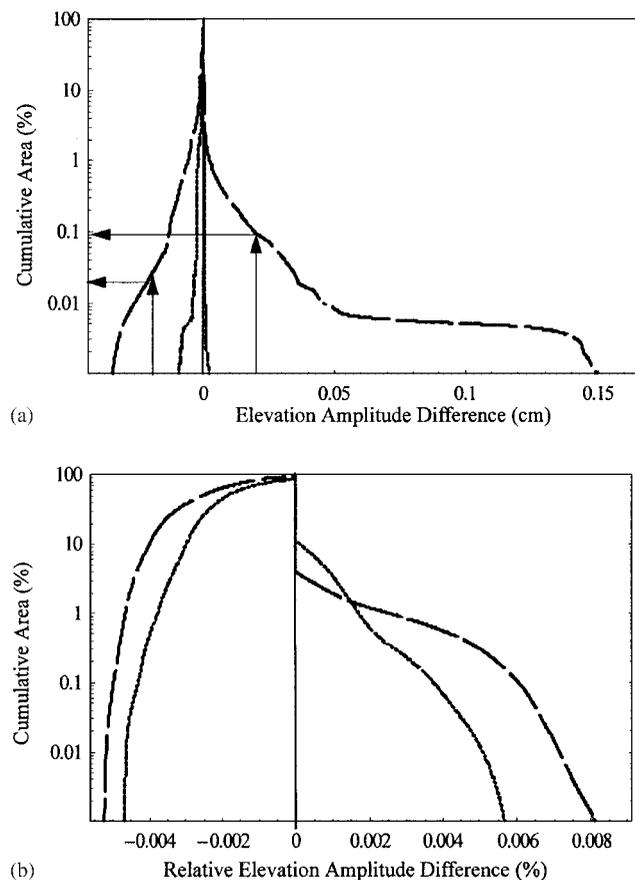


Figure 4. Temporal accuracy results for: (a) Bahamas, (b) Gulf of Mexico and (c) Eastcoast. All results are based on the  $L_2$  norm of elevation (dot-dash line—original algorithm (labelled OA), solid line—predictor–corrector algorithm (labelled PC)).

Table III. Order of accuracy from slope of  $L_2$  norm for the elevation field (velocity field shows similar results).

Domains	Original	Predictor–corrector
Quarter annular ( $30 \times 30$ )	1.15	2.03
Bahamas	1.27	1.89
Gulf of Mexico	1.19	1.97
Eastcoast	1.53	1.92

Figure 5. Sample CAFE plots. Vertical solid line at  $x=0$  represents a ‘perfect’ solution (no error).

then follows that value to where it intersects the CAFE plot and obtains the cumulative area that exceeds this convergence criteria (in this case, 0.095% overprediction and 0.02% underprediction). Note that in Figure 5(a), the short-dashed line plots underneath the curve of the long-dashed line, indicating less error over the entire domain. Also, by examining the shape of the CAFE plots, we can infer the spatial distribution of the temporal error, e.g. if the

Table IV. Elevation error measures for Eastcoast domain.

	Eastcoast (same $\Delta t$ )		Eastcoast (different $\Delta t$ )	
	Original	PC	Original	PC
Time step (s)	40	40	40	100
Elevation amplitude (absolute)				
% exceeding $-0.01$ cm	0.3	<b>0.001</b>	0.3	<b>0.03</b>
% exceeding $0.01$ cm	0.3	<b>&lt;0.0001</b>	0.3	<b>0.04</b>
% exceeding $\pm 0.01$ cm	0.6	<b>&lt;0.0011</b>	0.6	<b>0.07</b>
Elevation amplitude (relative)				
% exceeding $-0.1\%$	0.1	<b>0.0009</b>	0.1	<b>0.01</b>
% exceeding $0.1\%$	0.08	<b>0.007</b>	0.08	<b>0.004</b>
% exceeding $\pm 0.1\%$	0.18	<b>0.0079</b>	0.18	<b>0.014</b>
Elevation phase difference				
% exceeding $-0.1^\circ$	0.025	<b>&lt;0.0001</b>	0.025	<b>0.0008</b>
% exceeding $0.1^\circ$	0.02	<b>0.003</b>	0.02	<b>0.002</b>
% exceeding $\pm 0.1^\circ$	0.045	<b>&lt;0.0031</b>	0.045	<b>0.0028</b>

curves are ‘skinny’ (narrow at the top), the temporal error is confined to a small portion of the domain, but if the curves are ‘fat’ (wider at the top—as shown in Figure 5(b)), temporal error permeates the domain.

Tables IV–VII present the error levels obtained from the CAFE results for two domains: Eastcoast and Gulf of Mexico ( $\lambda/\Delta x$ ). Results are representative of the accuracy changes for all the domains. These tables show the error levels for all of the velocity and elevation components for the indicated temporal resolutions. The criteria used in these tables are based on tolerance levels that exceed required accuracy for most applications. In the tables, the bold values indicate the lowest error. In analysing the results shown in these tables, we observe that when the time step is the same between the two algorithms, the predictor–corrector algorithm *always* produces less error. (Note that all bold numbers are under the ‘PC, same  $\Delta t$ ’ column.) Noteworthy is that the error measure is often two orders of magnitude less. When the time step for the predictor–corrector algorithm is 2.5 times that of the original algorithm for Eastcoast and 3 times that of the original algorithm for the Gulf of Mexico, the results show the predictor–corrector algorithm still produces less error than the original algorithm at a lower time step in most ( $\approx 81\%$ ) of the cases. (Note that the majority of bold numbers are still under the ‘PC, different  $\Delta t$ ’ column.) Moreover, many of these predictor–corrector errors for the Eastcoast domain are still an order of magnitude less than the original algorithm.

#### 5.4. Testing of the combined parallel/predictor–corrector algorithm

Complex applications on larger, more intricate domains require that we utilize parallel computing to obtain results in a time-efficient manner. Therefore, the last component of this work with the predictor–corrector algorithm was to code and analyse the algorithm in parallel. The original parallel code is reported in Dawson *et al.* [24], and it uses the METIS algorithm [25] to decompose grids. Ghost nodes and MPI (message passing interface) are used to pass information between the subdomains at each iteration. Further information on the parallel code

Table V. Velocity error measures for Eastcoast domain.

	Eastcoast (same $\Delta t$ )		Eastcoast (different $\Delta t$ )	
	Original	PC	Original	PC
Time step (s)	40	40	40	100
Major semi-axis (absolute)				
% exceeding $-0.01 \text{ cm s}^{-1}$	0.2	< <b>0.0001</b>	0.2	< <b>0.0001</b>
% exceeding $0.01 \text{ cm s}^{-1}$	0.04	< <b>0.0001</b>	0.04	< <b>0.0001</b>
% exceeding $\pm 0.01 \text{ cm s}^{-1}$	0.24	< <b>0.0002</b>	0.24	< <b>0.0002</b>
Major semi-axis (relative)				
% exceeding $-0.1\%$	0.08	<b>0.0009</b>	0.08	<b>0.002</b>
% exceeding $0.1\%$	0.003	<b>0.002</b>	<b>0.003</b>	0.006
% exceeding $\pm 0.1\%$	0.083	<b>0.0029</b>	0.083	<b>0.008</b>
Major semi-axis phase difference				
% exceeding $-0.05^\circ$	0.14	<b>0.003</b>	0.14	<b>0.04</b>
% exceeding $0.05^\circ$	0.9	<b>0.001</b>	0.9	<b>0.04</b>
% exceeding $\pm 0.05^\circ$	1.04	<b>0.004</b>	1.04	<b>0.08</b>
Eccentricity				
% exceeding $-0.0002$	1	<b>0.015</b>	1	<b>0.8</b>
% exceeding $0.0002$	2	<b>0.012</b>	2	<b>0.4</b>
% exceeding $\pm 0.0002$	3	<b>0.027</b>	3	<b>1.2</b>
Major semi-axis direction				
% exceeding $-0.1^\circ$	0.07	< <b>0.0001</b>	0.007	<b>0.0005</b>
% exceeding $0.1^\circ$	0.016	< <b>0.0001</b>	<b>0.0016</b>	0.004
% exceeding $\pm 0.1^\circ$	0.086	< <b>0.0002</b>	0.0086	<b>0.0045</b>

Table VI. Elevation error measures for Gulf of Mexico domain.

	Gulf of Mexico (same $\Delta t$ )		Gulf of Mexico (different $\Delta t$ )	
	Original	PC	Original	PC
Time step (s)	50	50	50	150
Elevation amplitude (absolute)				
% exceeding $-0.002 \text{ cm}$	4	< <b>0.0001</b>	4	<b>3</b>
% exceeding $0.002 \text{ cm}$	2	< <b>0.0001</b>	2	<b>0.7</b>
% exceeding $\pm 0.002 \text{ cm}$	6	< <b>0.0002</b>	6	<b>3.7</b>
Elevation amplitude (relative)				
% exceeding $-0.1\%$	0.4	<b>0.007</b>	0.4	<b>0.25</b>
% exceeding $0.1\%$	0.15	<b>0.02</b>	<b>0.15</b>	0.4
% exceeding $\pm 0.1\%$	0.55	<b>0.027</b>	<b>0.55</b>	0.65
Elevation phase difference				
% exceeding $-0.1^\circ$	0.07	<b>0.015</b>	<b>0.07</b>	0.1
% exceeding $0.1^\circ$	0.08	< <b>0.0001</b>	<b>0.08</b>	0.1
% exceeding $\pm 0.1^\circ$	0.15	< <b>0.0151</b>	<b>0.15</b>	0.2

can be found in Dawson *et al.* [24]. We adapted this strategy for the combined predictor–corrector/parallel code.

Figure 6 shows an example of the decomposition of the quarter annular grid ( $100 \times 100$  resolution) on four processors. Figure 7 shows the surface to volume ratio (computed as

Table VII. Velocity error measures for Gulf of Mexico domain.

	Gulf of Mexico (same $\Delta t$ )		Gulf of Mexico (different $\Delta t$ )	
	Original	PC	Original	PC
Time step (s)	50	50	50	150
Major semi-axis (absolute)				
% exceeding $-0.001 \text{ cm s}^{-1}$	2.5	<b>0.08</b>	<b>2.5</b>	6
% exceeding $0.001 \text{ cm s}^{-1}$	3.5	<b>&lt;0.0001</b>	3.5	<b>0.04</b>
% exceeding $\pm 0.001 \text{ cm s}^{-1}$	6	<b>&lt;0.0801</b>	<b>6</b>	6.04
Major semi-axis (relative)				
% exceeding $-0.02\%$	60	<b>0.005</b>	60	<b>5</b>
% exceeding $0.02\%$	2	<b>0.01</b>	<b>2</b>	5
% exceeding $\pm 0.02\%$	62	<b>0.015</b>	62	<b>10</b>
Major semi-axis phase difference				
% exceeding $-0.02^\circ$	2	<b>0.002</b>	<b>2</b>	8
% exceeding $0.02^\circ$	10	<b>0.022</b>	10	<b>1.5</b>
% exceeding $\pm 0.02^\circ$	12	<b>0.024</b>	12	<b>9.5</b>
Eccentricity				
% exceeding $-0.0002$	3.5	<b>&lt;0.0001</b>	3.5	<b>1.5</b>
% exceeding $0.0002$	4	<b>&lt;0.0001</b>	4	<b>1.25</b>
% exceeding $\pm 0.0002$	7.5	<b>&lt;0.0002</b>	7.5	<b>2.75</b>
Major semi-axis direction				
% exceeding $-0.02^\circ$	2	<b>0.0004</b>	2	<b>0.8</b>
% exceeding $0.02^\circ$	1.5	<b>&lt;0.0001</b>	<b>1.5</b>	2
% exceeding $\pm 0.02^\circ$	3.5	<b>&lt;0.0005</b>	3.5	<b>2.8</b>

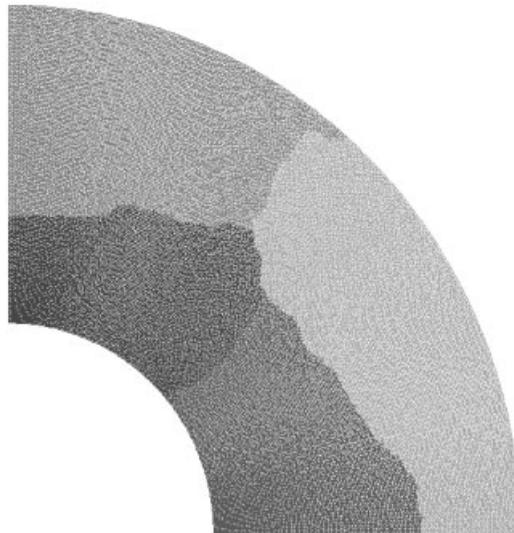


Figure 6. Example of the domain decomposition. The quarter annular domain with 100 000 nodes is shown for a four processor decomposition.

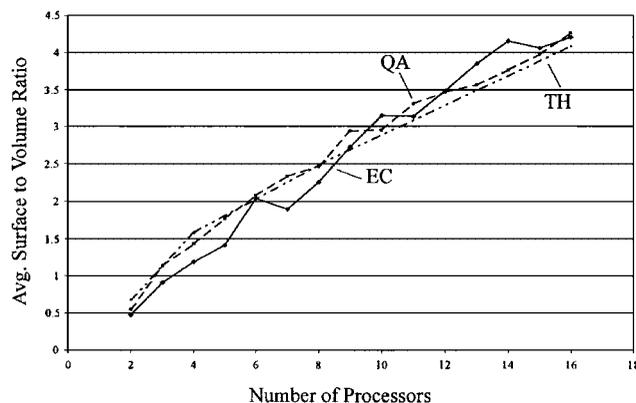


Figure 7. Surface to volume ratio vs number of subdomains. Dashed line—quarter annular domain ( $100 \times 100$  resolution)—labelled QA, Solid line with diamonds—Eastcoast domain—labelled EC. Theoretical values are shown with the dashed-dot-dot line—labelled TH.

the ratio of the number of nodes on the boundary of the subdomain to the total number of nodes in the subdomain) versus the number of processors for two applications discussed herein, quarter annular and Eastcoast. For computational efficiency, this ratio should be kept as low as possible in order to keep communication costs low. Here, for the quarter annular domain, an ideal surface to volume ratio can be computed because it is topologically similar to a square, which, of all rectangular figures, is the one that has the maximum area for given perimeter. Thus, if a larger square is divided so that each subdomain is square, it will maintain the minimum possible surface to volume ratio. Such a computation for a square with the same number of nodes as the quarter annular domain is shown as the dashed-dot-dot line in Figure 7. As can be seen, the actual surface to volume ratio for the quarter annular domain (see the dashed line in Figure 7) nearly matches the ideal curve for the square. Surprisingly, the Eastcoast domain also shows very good surface to volume behaviour even though it is a very irregular domain (see the solid line in Figure 7). Thus, the METIS algorithm appears to be producing near-optimal subdomains.

*5.4.1. Methodology and domains evaluated for the benchmarking studies.* The parallel code was benchmarked on two platforms. Table VIII describes the characteristics of the two parallel computing environments, which are two 16-processor clusters: one consisting of Sun UltraSparc IIe processors, and another consisting of Intel Pentium III processors.

In order to determine the performance of the new combined algorithm, we investigated three scenarios, which are presented in Table IX. Two scenarios (quarter annular- $100 \times 100$  resolution and Eastcoast domains) provide information on scaling and the effects of the additional stability associated with the predictor-corrector algorithm. In the third study, we want to minimize the impact of communication overhead on scaling by keeping the number of nodes per processor constant, as indicated in the table. Under this scenario, the computational workload per processor is kept constant.

Table VIII. Comparison of the two computer architectures.

Attributes	Sun Ultra Sparc IIe	Intel Pentium III
Speed	500 MHz	1 GHz
Operating system	Solaris 8	Linux
Cache	256 MB	256 MB
Memory	128 MB	256 MB
Communication	100 Mb/s	100 Mb/s
Compiler	Sun Forte 6.0	NAG
MPI	Sun ClusterTools	MPIch

Table IX. Benchmarking studies—parameters.

Parameters/ domains [study]	Quarter annular (100 × 100) [1]	Eastcoast [1]	Quarter annular [2]
Number of nodes	100 000	32 947	5000–80 000
Avg. number of nodes/processor	varies	varies	5000
Time step (original)	30	60	25
Time step (predictor–corrector)	180	515	25

*5.4.2. Results of the benchmarking studies.* In this section, we focus on two main comparisons: (1) results when the global number of nodes remains constant; and (2) results when the average number of local nodes per processor remains constant. Results from the first study are shown in Figure 8. Figure 8(a) shows the actual run time information and Figure 8(b) shows the scaling results for the quarter annular domain. For the timing information, wall-clock times were obtained for both algorithms at the time step indicated in Table IX for the Sun and Intel platforms. To develop the scaling results, we compared the timing results from each processor to the two processor simulation, which served as the baseline. In theory, the four processor simulation should run twice as fast as the two processor simulation, etc.; this theoretical speed-up is shown in Figure 8(b) as a solid line. From these figures, we observe the following:

- As expected, the parallel version of the predictor-corrector algorithm also shows significant stability gains. For example, with the quarter annular domain, we found a six-fold increase in the maximum time step with both serial and parallel versions.
- Results for both the Sun and Intel platforms indicate the predictor–corrector algorithm significantly reduces wall-clock time because of the gains in stability. Comparing the two platforms, we see that Intel results show a slightly faster wall-clock time as compared to the Sun results, which is most likely due to the higher clock speed of the Intel processors.
- Scalability results for both platforms show a near linear speed-up through the six-processor simulation, with a slight decrease as the number of processor increases. For both algorithms, we find the same trend with the slight decrease at the higher number of processors being more pronounced for the predictor–corrector algorithm.
- We also evaluated the Eastcoast domain (figures not shown) with the combined parallel, predictor–corrector algorithm and observed similar behaviour, e.g. an eight-fold increase

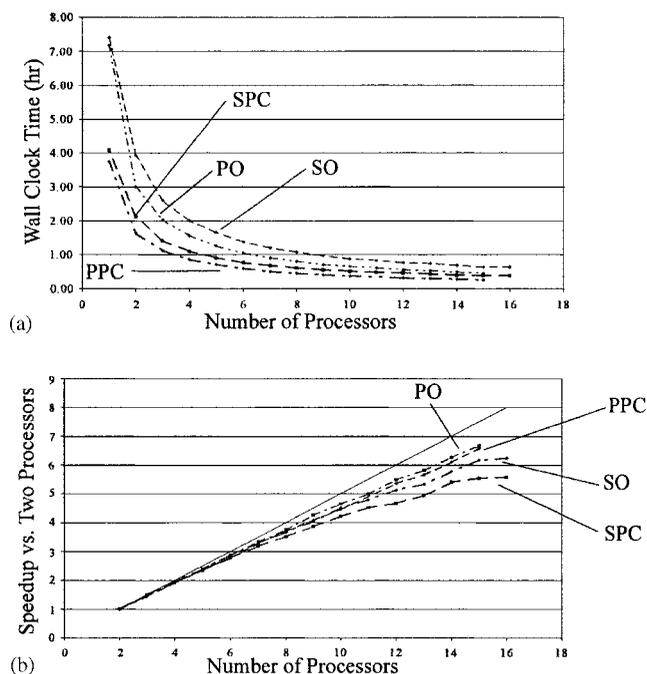


Figure 8. Benchmarking results for the quarter annular scenario with (a) showing the wall-clock time while (b) shows the speed-up vs two processors. (PO—original time-marching algorithm (Pentium), PPC—predictor–corrector time-marching algorithm (Pentium), SO—original time-marching algorithm (Sun) and SPC—predictor–corrector time-marching algorithm (Sun)). Solid line in (b) indicates the theoretical speed-up (linear).

in the maximum time step with both serial and parallel versions, good scaling, and lower wall-clock time for the predictor–corrector algorithm.

Figure 9 shows the timing results for the third study, where the quarter annular domain is used and the average number of nodes per processor is held constant. Significant differences in wall-clock time between the two algorithms exist because we use the same time step for both algorithms (recall the predictor–corrector algorithm requires twice the computation time per time step, thus causing the simulation to take approximately twice as long). In this study, the wall-clock time should theoretically remain constant as the number of processors increases because the average workload per processor remains constant. Results indicate the Sun platform has a significant increase in the wall-clock time from one to two processors, which does not occur for the Intel platform. This deviation from the constant theoretical results indicates communication overhead is greater for the Sun platform than for the Intel platform. For the Sun platform, wall-clock time continues to increase through the ten processor simulation, then levels out. For the Intel platform, the wall-clock times show less increase as the number of processors increases. Differences between the two platforms may be due to the communication configurations and MPI implementation.

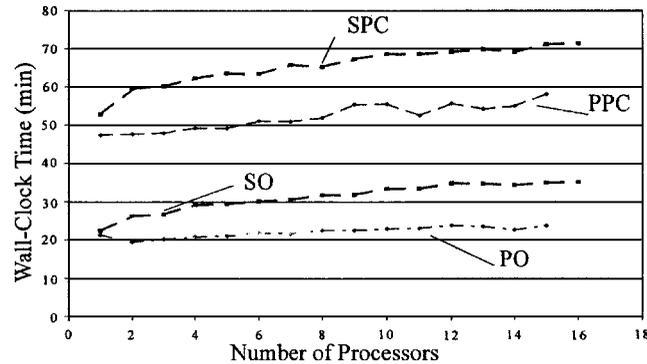


Figure 9. Benchmarking study results for the two time-marching algorithms where the time step is constant between the two algorithms in the quarter annular domain. (PO—original time-marching algorithm (Pentium), PPC—predictor–corrector time-marching algorithm (Pentium), SO—original time-marching algorithm (Sun) and SPC—predictor–corrector time-marching algorithm (Sun)).

## 6. CONCLUSIONS

In this paper, we develop and analyse a predictor–corrector algorithm for 2D, GWCE-based shallow water models. We quantify the effects of this new time-marching algorithm with respect to the stability and temporal accuracy (both globally and locally) for a wide variety of 2D domains and also look at the influence of mesh generating techniques ( $\lambda/\Delta x$  versus LTEA) on the results for the Gulf of Mexico domain. Lastly, we evaluate the computational performance of the combined parallel/predictor–corrector algorithm, as compared to the original algorithm. The hypothesis put forth in Kolar *et al.* [5] suggest that the stability constraint stems primarily from the explicit evaluation of non-linear terms. From the results presented in this paper, it is evident that the stability constraint is indeed relaxed with the implicit evaluation of the non-linear terms, therefore confirming this hypothesis. Other major findings from this 2D study are listed below.

- With all of the non-linear terms treated implicitly, stability shows dramatic improvement, ranging from a minimum of a three-fold increase with the Gulf of Mexico domain ( $\lambda/\Delta x$  resolution) to a maximum of an eight-fold increase with the Eastcoast domain. As mentioned earlier, for this algorithm to be considered cost-effective, we must obtain at least a 100% change between the two algorithms (assuming one iteration of the corrector stage).
- Mesh generating techniques influence the allowable Courant number in the Gulf of Mexico for both algorithms. In particular, the LTEA mesh minimizes the spatial variability of the Courant number over the shelf break region, which coincides with the area where the extra refinement occurs in this mesh. The finer resolution produces a smaller allowable Courant number, but allows a larger percent increase from the original to the predictor–corrector.
- Resolution studies for the quarter annular domain show that the allowable Courant number with the original algorithm remains constant with increasing resolution; however, allowable Courant numbers increase with resolution for the predictor–corrector algorithm.

- From the  $G$  sensitivity study, we see that the  $G$  values that produce minimal mass balance errors and errors in the generation of the non-linear constituents coincide with those that allow the maximum stable time step (i.e.  $1 \leq G/\tau_{\max} \leq 10$ ).
- Global temporal accuracy (convergence rate) studies show that the predictor–corrector algorithm reduces absolute error and increases the order of accuracy from approximately first order to nearly second order.
- From the local temporal accuracy studies, we determined the predictor–corrector algorithm decreases errors by approximately two orders of magnitude, as compared to the original algorithm at the same time step.
- When evaluating the predictor–corrector algorithm with a time step that is at least twice that of the original algorithm, we found that the predictor–corrector algorithm still produces less error in most of the components (approximately 81%) than the original algorithm.
- Results from the combined parallel/predictor–corrector algorithm show that it significantly reduces simulation times as compared to the parallel algorithm that does not use predictor–corrector time marching. Thus, the enhanced stability allows us to produce quick results for time-sensitive applications.
- For both computing platforms (Intel and Sun) the combined algorithm achieves nearly ideal speedup through six processors, with slight tapering-off as the number of processors is increased. More importantly, when the workload per processor is kept nearly constant, the scaling at a higher number of processors is more ideal.

## NOMENCLATURE

<b>A</b>	atmospheric force ( $L^2/T^2$ )
$C_r$	Courant number, equals $c\Delta t/\Delta x$
$C^0$	set of continuous functions over $\Omega$ whose first derivative is, at most, discontinuous at a finite number of points in $\Omega$
$G$	numerical parameter in the generalized wave continuity equation ( $1/T$ )
$H$	total fluid depth, equals $h + \zeta(L)$
$L$	symbol for primitive continuity equation
<b>M</b>	symbol for primitive momentum equation, non-conservative form
<b>M<sup>c</sup></b>	symbol for primitive momentum equation, conservative form
<b>T</b>	macroscopic stress tensor ( $M/LT^2$ )
$W^G$	symbol for the generalized wave continuity equation
$c$	linear wave celerity, equals $\sqrt{gh}$
$f$	Coriolis parameter, equals $2\Omega \sin \phi$
$g$	gravity $ \mathbf{g} $ ( $L/T^2$ )
$h$	bathymetry ( $L$ )
$k$	time weighting parameter or temporal index
$p$	pressure ( $M/LT^2$ )
$p_a$	atmospheric pressure ( $M/LT^2$ )
$t$	time
<b>v</b>	depth-averaged velocity of the fluid ( $L/T$ )

*Greek letters*

$\Omega$	angular velocity of the earth ( $1/T$ )
$\alpha$	Earth elasticity factor
$\varepsilon$	eddy viscosity ( $L^2/T$ )
$\zeta$	elevation of water surface above the datum ( $L$ )
$\eta$	Newtonian equilibrium tidal potential
$\lambda$	wavelength
$\rho$	density ( $M/V$ )
$\tau$	linearized bottom friction ( $1/T$ )

*Special symbols and operators*

$\nabla$	nabla (grad) operator in $x$ - $y$ plane ( $1/L$ )
$\nabla \cdot$	divergence operator in $x$ - $y$ plane ( $1/L$ )
$\partial/\partial t$	partial derivative ( $1/T$ )

## ACKNOWLEDGEMENTS

Financial support for this research was provided in part by the National Science Foundation under the contracts ACI-9623592 and EEC-9912319 and the United States Department of Education GAANN Fellowship. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding agencies. We would also like to thank Henry Neeman, director of OSCER (OU Supercomputing Center for Education and Research) for his contributions. Lastly, we thank the anonymous reviewers for their constructive comments.

## REFERENCES

1. Dresback KM, Kolar RL. Implicit time-marching algorithm for shallow water models based on the generalized wave continuity equation. *International Journal for Numerical Methods in Fluids* 2001; **36**:925–945.
2. Lynch DR, Gray WG. A wave equation model for finite element tidal computations. *Computers and Fluids* 1979; **7**(3):207–228.
3. Kinnmark IPE. The shallow water wave equations: formulations, analysis and application. In *Lecture Notes in Engineering*, Brebbia CA, Orszag SA (eds), vol. 15. Springer: Berlin, 1996; 1–187.
4. Luettich RA, Westerink JJ, Scheffner NW. ADCIRC: an advanced three-dimensional circulation model for shelves, coasts and estuaries. Report 1: theory and methodology of ADCIRC-2DDI and ADCIRC-3DL. *Technical Report DRP-92-6*, Department of the Army, USACE, Washington, DC, 1992.
5. Kolar RL, Looper JP, Westerink JJ, Gray WG. An improved time marching algorithm for GWC shallow water models. In *CMWR XII Volume 2: Computational Methods in Surface Flow and Transport Problems*, Burganos VNI, Karatzas GP, Payatakes AC, Brebbia CA, Gray WG, Pinder GF (eds). Computational Mechanics Publications: Southampton, 1998; 379–385.
6. Lee JK, Froehlich DC. *Review of literature on the finite-element solution of the equations of two-dimensional surface-water flow in the horizontal plane*. U.S. Geological Survey Circular 1009, USDOI, Denver, CO., 1986.
7. Gray WG, Lynch DR. Time-stepping schemes for finite element tidal model computations. *Advances in Water Resources* 1977; **1**(2):83–95.
8. Kinnmark IPE, Gray WG. An implicit wave equation model for the shallow water equations. *Advances in Water Resources* 1984; **7**:168–171.
9. Blain CA, Westerink JJ, Luettich RA. The influence of domain size on the response characteristics of a hurricane storm surge model. *Journal of Geophysical Research* 1994; **99**:18467–18479.
10. Blain CA, Westerink JJ, Luettich RA. Grid convergence studies for the prediction of hurricane storm surge. *International Journal for Numerical Methods in Fluids* 1998; **26**:369–401.
11. Hagen SC, Westerink JJ, Kolar RL. Two-dimensional, unstructured mesh generation for tidal models. *International Journal for Numerical Methods in Fluids* 2001; **35**:669–686.

12. Luettich RA, Westerink JJ. Implementation and testing of elemental flooding and drying in the ADCIRC hydrodynamic model. *Final Report*, 8/95, Dept. of the Army, USACE, Washington, DC, 1995.
13. Kolar RL, Gray WG, Westerink JJ. Boundary conditions in shallow water models—an alternative implementation for finite element codes. *International Journal for Numerical Methods in Fluids* 1996; **22**(7):603–618.
14. Lynch DR, Holboke MJ. Normal flow boundary conditions in 3D circulation models. *International Journal for Numerical Methods in Fluids* 1997; **25**(10):1185–1205.
15. Lynch DR, Werner FE, Greenberg DA, Loder JW. Diagnostic model for baroclinic and wind-driven circulation in shallow seas. *Continental Shelf Research* 1992; **12**(1):37–64.
16. Lynch DR. Three-dimensional diagnostic model for baroclinic, wind-driven and tidal circulation in shallow seas—FUNDY4 users' manual. *Numerical Methods Laboratory Report*, 1990.
17. Muccino JC, Gray WG, Foreman MGG. Calculation of vertical velocity in three-dimensional shallow water equation, finite element models. *International Journal for Numerical Methods in Fluids* 1997; **25**(7):779–802.
18. Chippada S, Dawson CN, Wheeler MF, Martinez ML. Parallel computing for finite element models of surface water flow. In *CMWR XI Volume 2: Computational Methods in Surface Flow and Transport Problems*, Aldama AA, Aparicio J, Brebbia CA, Gray WG, Herrera I, Pinder GF (eds). Computational Mechanics Publication: Southampton, 1996; 63–70.
19. Kashiyama K, Saito K, Yoshikawa S. Massively parallel finite element method for large scale computation of storm surge. In *CMWR XI Volume 2: Computational Methods in Surface Flow and Transport Problems*, Aldama AA, Aparicio J, Brebbia CA, Gray WG, Herrera I, Pinder GF (eds). Computational Mechanics Publication: Southampton, 1996; 79–86.
20. Lynch DR, Werner FE. Three-dimensional hydrodynamics on finite elements. Part II: non-linear time-stepping model. *International Journal for Numerical Methods in Fluids* 1991; **12**:507–533.
21. Kolar RL, Gray WG, Westerink JJ, Luettich RA. Shallow water modeling in spherical coordinates: equation formulation, numerical implementation and application. *Journal of Hydraulic Research* 1994; **32**(1):3–24.
22. Kolar RL, Westerink JJ, Cantekin ME, Blain CA. Aspects of non-linear simulations using shallow-water models based on the wave continuity equation. *Computers and Fluids* 1994; **23**:523–528.
23. Luettich RA, Westerink JJ. Continental shelf scale convergence studies with a tidal model. In *Quantitative Skill Assessment for Coastal Ocean Models, Coastal and Estuarine Studies*. Lynch DR, Davies AM (eds), vol. 47. American Geophysical Union: Washington, DC, 1995; 349–371.
24. Dawson CN, Parr VJ, Wheeler MF. Issues in parallel computation of flow and transport in surface waters. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, Arabnia HR (ed.), vol. 1. CSREA Press: Las Vegas, NV, 2000; 21–27.
25. Karypis G, Kumar V. METIS 4.0: unstructured graph partitioning and sparse matrix ordering system. *Technical Report*, Department of Computer Science and Engineering, University of Minnesota, 1998.