SIAM J. SCI. COMPUT.
Vol. 22, No. 6, pp. 2075–2092
© 2001 Society for Industrial and Applied Mathematics

# TWO ELEMENT-BY-ELEMENT ITERATIVE SOLUTIONS FOR SHALLOW WATER EQUATIONS*

C. C. FANG† AND TONY W. H. SHEU†

**Abstract.** In this paper we apply the generalized Taylor–Galerkin finite element model to simulate bore wave propagation in a domain of two dimensions. For stability and accuracy reasons, we generalize the model through the introduction of four free parameters. One set of parameters is rigorously determined to obtain the high-order finite element solution. The other set of free parameters is determined from the underlying discrete maximum principle to obtain the monotonic solutions. The resulting two models are used in combination through the flux correct transport technique of Zalesak, thereby constructing a finite element model which has the ability to capture hydraulic discontinuities. In addition, this paper highlights the implementation of two Krylov subspace iterative solvers, namely, the bi-conjugate gradient stabilized (Bi-CGSTAB) and the generalized minimum residual (GMRES) methods. For the sake of comparison, the multifrontal direct solver is also considered. The performance characteristics of the investigated solvers are assessed using results of a standard test widely used as a benchmark in hydraulic modeling. Based on numerical results, it is shown that the present finite element method can render the technique suitable for solving shallow water equations with sharply varying solution profiles. Also, the GMRES solver is shown to have a much better convergence rate than the Bi-CGSTAB solver, thereby saving much computing time compared to the multifrontal solver.

**Key words.** Taylor–Galerkin finite element model, discrete maximum principle, flux correct transport technique, Bi-CGSTAB, GMRES, multifrontal direct solver, sharply varying

**AMS subject classifications.** 65F10, 65N30, 76B15

**PII.** S1064827599360881

## 1. Introduction.

Many environmental problems, such as tides in oceans, breaking waves on shallow beaches, flood waves in rivers, mountain torrents, and estuary flows [1] are closely related to the motion of unsteady free-surface flow. Predicting the height and speed of the bore wave is the first step in providing useful information for flood control and for the design of channel walls. It is the practical importance of simulating shallow water equations that motivated the present study.

The shallow water height is analogous to gas density in gas dynamic equations. Since gas dynamic equations admit discontinuous solutions, called shocks and contact discontinuities, this analogy between two fields of equations implies that it is possible to observe hydraulic jumps and bores in water and in the atmosphere. Numerically capturing these discontinuous phenomena in hydraulics has become a major area of theoretical and computational study. For suppressing dispersive oscillations exhibited near the shock front, significant effort has been directed toward the development of high-resolution hydraulic methods. Shock-capturing methods were first developed by Godunov [2] and Van Leer [3]. Development of high-resolution schemes was followed by adoption of the total variation diminishing (TVD) scheme of Harten [4], the parabolic method (PPM) of Colella and Woodward [5], and the essentially nonoscillatory (ENO) schemes of Harten and Osher [6] to obtain numerically very accurate but computationally absolute stable solutions. Most of these high-resolution schemes

*Received by the editors September 2, 1999; accepted for publication (in revised form) August 23, 2000; published electronically March 28, 2001. This work was supported by National Science Council of Republic of China grant NSC 88-2611-E-002-025.
http://www.siam.org/journals/sisc/22-6/36088.html
†Department of Naval Architecture and Ocean Engineering, National Taiwan University, 73 Chou-Shan Rd., Taipei, Taiwan, Republic of China (Tony.Sheu@cfd.na.ntu.edu.tw).

were, unfortunately, developed within the one-dimensional framework. The desire to avoid this limitation has prompted many researchers to capture bore waves in an open-channel flow and hydraulic jumps in a genuinely multidimensional dam-break problem [7, 8, 9, 10, 11, 12, 13, 14, 15]. Since the flux corrected transport (FCT) algorithm was originally developed without resorting to a single spatial dimension [16, 17], we consider the FCT algorithm to be one of the most suitable choices for multidimensional hydraulic calculations.

As is typical with other finite element flow simulations, we work with a large matrix equation. Thus, we must minimize this disadvantage if we are to compete with other structured-type discretization methods. To this end, we consider in the present work two iterative solvers and one very effective direct solver. Both iterative solvers, namely, the bi-conjugate gradient stabilized (Bi-CGSTAB) of Van der Vorst [18] and the generalized minimum residual (GMRES) of Saad and Schultz [19], are the ensemble of conjugate gradient method for solving the non-Hermitian linear system of algebraic equations. These two Krylov subspace methods differ in the vectors used to iterate the approximation solution. GMRES iterates the approximate solution in terms of Arnoldi vectors while Bi-CGSTAB accomplishes the same task through the use of unsymmetric Lanczos vectors. These vectors are chosen to overcome the difficulty regarding the nonexistence of the orthogonal tridiagonalization of the finite element stiffness matrix for shallow water equations. Since we wish to address the performance of solution solvers, we also consider the direct solver for completeness. In order to compete with the above state-of-the-art iterative solvers, we consider in this assessment study the multifrontal direct solver of Duff and Reid [20]. This solver is regarded as a refinement of the frontal solver of Irons [21].

The remainder of this paper is organized in six sections. Section 2 presents assumptions that lead to the St. Venant shallow water equations. In section 3, we present the Taylor–Galerkin finite element model, which can faithfully preserve the inherent hyperbolic conservation law [22]. Specific to the present finite element model is that four free parameters and one damping parameter are used to obtain higher prediction accuracy in the high-order scheme while accommodating the monotonicity property in the low-order scheme [23]. Section 3.3 explains how the high-order and low-order Taylor–Galerkin formulations can be used in combination to obtain a nonoscillatory solution. This is followed by presentation of two iterative solution solvers and one direct solver that can be used to solve the nonsymmetric finite element equations. The objective is to assess the efficiency of the solution solvers used to obtain finite element solutions from indefinite and unsymmetric matrix equations. In section 5, we provide analytical evidence by showing that the numerical model and solution solvers can render techniques suitable for hydraulic problems with sharp gradients. All results of model tests are well in agreement with the analytic data. With this success in analytical validation, we proceed to study the dam-break problem. Finally, a brief discussion together with some conclusions is presented in section 6.

**2. Mathematical formulation.** Shallow water equations are derived under zero fluid viscosity and surface tension assumptions. Wind shear and Coriolis forces are not taken into account. Working equations for incompressible free-surface fluid flows are derived under the small bottom slope condition. Another key assumption in the derivation is that the vertical component of the flow acceleration has negligible influence on the pressure. A hydrostatic pressure distribution is thus assumed. Given the above assumptions, the St. Venant shallow water equations, which govern mass and momentum conservation, are derived in terms of a solution vector $\mathbf{U}$ of the

conservative type [24]

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}, \tag{2.1}$$

where $\mathbf{U} = (h, hu, hv)^{\mathrm{T}}$. In the above, $h$ is the water depth, and $u$ and $v$ are the depth-averaged velocity components in the $x$- and $y$-directions, respectively. Denote $g$ as the gravitational acceleration; the physical fluxes are derived as $\mathbf{F} = (h\,u, h\,u^2 + \frac{1}{2}g\,h^2, h\,u\,v)^{\mathrm{T}}$ and $\mathbf{G} = (h\,v, h\,u\,v, h\,v^2 + \frac{1}{2}g\,h^2)^{\mathrm{T}}$. Without loss of generality, both friction losses and bed slopes are neglected to simplify the analysis.

Given initially smooth data for the present homogeneous case ($\mathbf{S} = \mathbf{0}$), the quasi-linear hyperbolic system of partial differential equations may admit discontinuities, such as bore waves that are often observed in practice, owing to nonlinear advective terms in the equations [25]. In time-accurate simulation of St. Venant equations, it is customary to rewrite working equations in their nonconservative equivalent forms to better show the characteristic nature of the hyperbolic system. Transformation of the conservative form into its nonconservative counterpart involves using the gravity wave velocity $c = (g\,h)^{1/2}$. The resulting eigenvectors and eigenvalues are critical in hydraulic simulation since they represent the characteristic speed and direction of signal transmission.

**3. Finite element model.** Within the weighted residual context, (2.1) can be approximated in its weak form through the use of a test function $\mathbf{W}$. This leads to

$$\sum_{el=1}^{n_{el}} \int_{\Omega^{el}} \int_{t_n}^{t_{n+1}} \mathbf{W} \left[ \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right] d\,t\,d\,\Omega^{el} = 0. \tag{3.1}$$

Define $\delta\mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n$ and $\mathbf{I} = \frac{\partial}{\partial x}\int_{t_n}^{t_{n+1}} \mathbf{F}\,dt + \frac{\partial}{\partial y}\int_{t_n}^{t_{n+1}} \mathbf{G}\,dt$; (3.1) can be expressed as follows through time integration:

$$\sum_{el=1}^{n_{el}} \int_{\Omega^{el}} (\mathbf{W}\delta\mathbf{U}^n - \mathbf{W}\,\mathbf{I})\,d\Omega^{el} = 0. \tag{3.2}$$

Analysis is carried out by performing Taylor series expansion of $\mathbf{F}$ and $\mathbf{G}$ with respect to $t_n$. Take $\mathbf{F}$ as an example; we can represent this vector in terms of Taylor series expansion terms terminated at the time increment $(t - t_n)^3$:

$$\mathbf{F} = \mathbf{F}^n + \left.\frac{\partial \mathbf{F}}{\partial t}\right|^n (t - t_n) + \frac{1}{2}\left.\frac{\partial^2 \mathbf{F}}{\partial t^2}\right|^n (t - t_n)^2 + \mathcal{O}(t - t_n)^3. \tag{3.3}$$

Recall that $\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{F}_x - \mathbf{G}_y$ and $\frac{\partial \mathbf{F}}{\partial t} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial t} = \mathbf{A}\frac{\partial \mathbf{U}}{\partial t}$; we introduce two free parameters $\alpha$ and $\beta$ and rewrite $\partial\mathbf{F}/\partial t$ exactly as $\frac{\partial \mathbf{F}}{\partial t} = \alpha\mathbf{A}\frac{\partial \mathbf{U}}{\partial t} + \beta\mathbf{A}[-\frac{\partial \mathbf{F}}{\partial x} - \frac{\partial \mathbf{G}}{\partial y}]$, provided that $\alpha$ and $\beta$ are constrained by $\alpha + \beta = 1$. Moreover, we can approximate the time derivative term $\partial\mathbf{F}/\partial t$ to obtain

$$\begin{aligned} \frac{\partial^2 \mathbf{F}}{\partial t^2} = &-\gamma \left( \mathbf{A}^2 \frac{\partial^2 \mathbf{U}}{\partial t \partial x} + \mathbf{A}\,\mathbf{B} \frac{\partial^2 \mathbf{U}}{\partial t \partial y} \right)\Bigg|^n \\ &+ \mu \left[ \mathbf{A}^2 \left( \frac{\partial^2 \mathbf{F}}{\partial x^2} + \frac{\partial^2 \mathbf{G}}{\partial x \partial y} \right) + \mathbf{A}\,\mathbf{B} \left( \frac{\partial^2 \mathbf{F}}{\partial x \partial y} + \frac{\partial^2 \mathbf{G}}{\partial y^2} \right) \right]\Bigg|^n . \end{aligned} \tag{3.4}$$

As in the above, the free parameters $\gamma$ and $\mu$, which are constrained by $\gamma + \mu = 1$, are also introduced. Substitution of $\frac{\partial \mathbf{F}}{\partial t}$ and $\frac{\partial^2 \mathbf{F}}{\partial t^2}$ into (3.3) leads to

$$\begin{aligned}\mathbf{F} = \mathbf{F}^n &+ \left[\alpha \mathbf{A} \frac{\partial \mathbf{U}}{\partial t} - \beta \mathbf{A} \left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y}\right)\right]\Bigg|^n (t - t_n) \\ &- \frac{1}{2}\left\{\gamma\left(\mathbf{A}^2 \frac{\partial^2 \mathbf{U}}{\partial t \partial x} + \mathbf{AB}\frac{\partial^2 \mathbf{U}}{\partial t \partial y}\right) - \mu\left[\mathbf{A}^2\left(\frac{\partial^2 \mathbf{F}}{\partial x^2} + \frac{\partial^2 \mathbf{G}}{\partial x \partial y}\right)\right.\right. \\ &\left.\left. + \mathbf{AB}\left(\frac{\partial^2 \mathbf{F}}{\partial x \partial y} + \frac{\partial^2 \mathbf{G}}{\partial y^2}\right)\right]\right\}\Bigg|^n (t - t_n)^2 + \mathcal{O}((t - t_n)^3). \qquad (3.5)\end{aligned}$$

Similarly, we can expand $\mathbf{G}$ with respect to quantities evaluated at time $t_n$:

$$\begin{aligned}\mathbf{G} = \mathbf{G}^n &+ \left[\alpha \mathbf{B} \frac{\partial \mathbf{U}}{\partial t} - \beta \mathbf{B} \left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y}\right)\right]\Bigg|^n (t - t_n) \\ &- \frac{1}{2}\left\{\gamma\left(\mathbf{BA} \frac{\partial^2 \mathbf{U}}{\partial t \partial x} + \mathbf{B}^2\frac{\partial^2 \mathbf{U}}{\partial t \partial y}\right) - \mu\left[\mathbf{BA}\left(\frac{\partial^2 \mathbf{F}}{\partial x^2} + \frac{\partial^2 \mathbf{G}}{\partial x \partial y}\right)\right.\right. \\ &\left.\left. + \mathbf{B}^2\left(\frac{\partial^2 \mathbf{F}}{\partial x \partial y} + \frac{\partial^2 \mathbf{G}}{\partial y^2}\right)\right]\right\}\Bigg|^n (t - t_n)^2 + \mathcal{O}((t - t_n)^3). \qquad (3.6)\end{aligned}$$

By substituting (3.5)–(3.6) into (3.2) and choosing bilinear polynomials as test and basis functions, we can derive the finite element equation in the $\delta$-form as follows:

$$\mathbf{M}_c\, \delta \mathbf{U}^n = \mathbf{R}. \qquad (3.7)$$

In the above, $\mathbf{M}_c$ denotes the consistent mass matrix:

$$\begin{aligned}\mathbf{M}_{ij}^{el} = \int_{\mathbf{\Omega}^{el}} &\left\{\mathbf{N}_i \mathbf{N}_j - \frac{1}{2}\alpha \Delta t \left(\frac{\partial \mathbf{N}_i}{\partial x} A + \frac{\partial \mathbf{N}_i}{\partial y} B\right)\mathbf{N}_j\right. \\ &+ \frac{1}{6}\gamma \Delta t^2 \left[\frac{\partial \mathbf{N}_i}{\partial x}\left(A^2 \frac{\partial \mathbf{N}_j}{\partial x} + AB \frac{\partial \mathbf{N}_j}{\partial y}\right) + \frac{\partial \mathbf{N}_i}{\partial y}\left(BA \frac{\partial \mathbf{N}_j}{\partial x} + B^2 \frac{\partial \mathbf{N}_j}{\partial y}\right)\right]\Bigg\}\, d\mathbf{\Omega}^{el} \\ - \int_{\mathbf{\Gamma}} &\left\{-\frac{1}{2}\alpha \Delta t \mathbf{N}_i \left(n_x A + n_y B\right)\mathbf{N}_j + \frac{1}{6}\gamma \Delta t^2 \mathbf{N}_i \left[n_x \left(A^2 \frac{\partial \mathbf{N}_j}{\partial x} + AB \frac{\partial \mathbf{N}_j}{\partial y}\right)\right.\right. \\ &\left.\left. + n_y \left(BA \frac{\partial \mathbf{N}_j}{\partial x} + B^2 \frac{\partial \mathbf{N}_j}{\partial y}\right)\right]\right\} d\mathbf{\Gamma}. \qquad (3.8)\end{aligned}$$

In (3.7), $\delta \mathbf{U}^n (\equiv \mathbf{U}^{n+1} - \mathbf{U}^n)$ is the vector of nodal increment and $\mathbf{R}$ $(\equiv \mathbf{C}\mathbf{F}^n + \tilde{\mathbf{C}}\mathbf{G}^n)$ is the vector of element contributions added to the finite element nodes. For detailed expressions of $\mathbf{C}$ and $\tilde{\mathbf{C}}$, refer to [23].

**3.1. High-order Taylor–Galerkin finite element model.** To resolve discontinuous solutions, we employ the FCT technique of Zalesak [17]. The idea behind this algorithm is to combine an accurate high-order scheme with a monotonic low-order scheme. To make this scheme effective, we require that the former scheme be used in the smooth regime and the low-order scheme only in regions near discontinuities.

The key to constructing an efficient FCT finite element model is to develop a model that can provide a high level of accuracy. To this end, we exploit the modified equation analysis for the determination of the free parameters $\alpha$, $\beta$, $\gamma$, and $\mu$ a priori to obtain higher-order accuracy. The strategy we adopt to achieve this goal is to take into consideration the scalar transport equation, $\phi_t + a\,\phi_x + b\,\phi_y = 0$, in the flow

with the constant velocity vector $\vec{u} = (a, b)$. The modified equation analysis reveals the rational use of $(\alpha^h,\ \beta^h,\ \gamma^h,\ \mu^h) = (0,\ 1,\ 1,\ 0)$ [23]. With these parameters, the resulting modified equation reads as

$$\begin{aligned} &\phi_t + a\ \phi_x + b\ \phi_y \\ &= T_1\,\phi_{xxxx} + T_2\,\phi_{xxxy} + T_3\,\phi_{xxyy} + T_4\,\phi_{xyyy} + T_5\,\phi_{yyyy} + \cdots, \end{aligned} \tag{3.9}$$

where $T_1 = \frac{1}{24}\,a\,\Delta x^3\,\nu_x\,({\nu_x}^2 - 1)$, $T_2 = \frac{1}{6}\,b\,\Delta x^3\,{\nu_x}^3$, $T_3 = -\frac{1}{12}\,b^2\,\Delta x^2\,\Delta t\,{\nu_x}^2$, $T_4 = \frac{1}{6}\,a\,\Delta y^3\,{\nu_y}^3$, $T_5 = \frac{1}{24}\,b\,\Delta y^3\,\nu_y\,({\nu_y}^2 - 1)$. In light of the spatial third-order accuracy, and the first-order temporal accuracy, the above Taylor–Galerkin finite element model shows promise as a means of predicting a smoothly distributed water height in shallow water equations.

**3.2. Low-order Taylor–Galerkin finite element model.** The next step in the development of the Taylor–Galerkin FCT (TG-FCT) finite element model is to derive the low-order model from the generalized Taylor–Galerkin finite element model. To achieve this goal, the model is not allowed to produce any nonphysical or numerical wiggles. This monotonicity and strictly positive field variable requirement is a key to success in any FCT method. The better the low-order scheme, the easier the task of limiting fluxes.

The development of a low-order finite element model proceeds as follows. We first rewrite (3.7) as

$$\mathbf{M}_c\,\mathbf{U}^{n+1} = \mathbf{R}^n + \mathbf{M}_c\mathbf{U}^n. \tag{3.10}$$

The derivation is followed by lumping the above equation to get

$$\mathbf{M}_l\,\mathbf{U}^{n+1} = \mathbf{R}^n + \mathbf{M}_c\mathbf{U}^n. \tag{3.11}$$

The above lumping-mass approximation helps to stabilize the discretized equation. Subtracting $\mathbf{M}_l\,\mathbf{U}^n$ from both sides of (3.11), we obtain

$$\mathbf{M}_l\,\delta\mathbf{U}^n = \mathbf{R}^n + (\mathbf{M}_c - \mathbf{M}_l)\,\mathbf{U}^n. \tag{3.12}$$

Further refinement of (3.12) can be made by multiplying $c_d$ $(0 \le c_d \le 1)$ by the added mass diffusion term to better control the predicted solution. This helps us avoid the introduction of unnecessarily large diffusion errors. The resulting model for obtaining the lower-order Taylor–Galerkin finite element solution $\mathbf{U}^n$ reads as

$$\mathbf{M}_l\,\delta\mathbf{U}^n = \mathbf{R}^n + c_d\,(\mathbf{M}_c - \mathbf{M}_l)\,\mathbf{U}^n, \tag{3.13}$$

where $\mathbf{M}_l = \mathcal{A}_{el=1}^{n_{el}}(\mathrm{diag}(\sum_{j=1}^{n_{\max}} \mathbf{M}_{cij}^{el}))$. In order to satisfy the requirement placed on the low-order scheme in any FCT method, we employ the discrete maximum theory [26, 27, 28]. Based on this underlying theory, the matrices involved in (3.13) are of the M-matrix type provided that the five free parameters introduced into the scalar formulation are prescribed as $(\alpha^l,\ \beta^l,\ \gamma^l,\ \mu^l,\ c_d) = (0,\ 0,\ 0,\ 0,\ 0.425)$.

When simulating nonlinear shallow water equations, we may encounter a sonic flow situation. In this case, entropy fix must be invoked in order to avoid nonphysical rarefaction shocks at the sonic point. To satisfy the entropy satisfaction property when the sonic condition is detected, we should add an entropy flux term, $\frac{\partial}{\partial x}(b(\nu_x)\frac{\partial \mathbf{U}}{\partial x}) + \frac{\partial}{\partial y}(b(\nu_y)\frac{\partial \mathbf{U}}{\partial y})$, to the region where it is needed. The damping coefficients used in the entropy flux are as follows [29]:

$$b(\nu_i) = c_e\ \frac{\Delta t}{2\lambda^2}\ q(\nu_i), \tag{3.14}$$

where $\lambda = \Delta t/\Delta x$ (or $\Delta t/\Delta y$), $\nu_i = \frac{\Delta t}{\Delta x}(u-c)$ (or $\nu_i = \frac{\Delta t}{\Delta y}(v-c)$), and

$$q(\nu_i) = \begin{cases} 0, & |\nu_i| \geq \epsilon, \\ \epsilon^2 - \nu_i^2, & |\nu_i| < \epsilon. \end{cases} \tag{3.15}$$

In what follows, we set $\epsilon = 0.2$ and $c_e = 2.0$.

**3.3. FCT filtering algorithm.** Having determined the values $(\alpha, \beta, \gamma, \mu, c_d)$ needed to obtain high- and low-order finite element solutions, we can use two Taylor–Galerkin models in combination to obtain positive and accurate results which are free of nonphysical fluctuations. Use of the two schemes in combination was proposed by Zalesak [17]. We follow closely the FCT scheme of Zalesak by calculating $\delta\mathbf{U}^h$ from the high-order model using either the iterative solvers or the multifrontal direct solver [30] discussed below.

Upon obtaining the solution $\delta\mathbf{U}^h$, we can compute the antidiffusive flux array $\mathbf{F}^{el^h}$ in each element:

$$\mathbf{F}^{el^h} = [F_i^{el^h}] = \mathbf{M}_{l_h}^{-1}\,[\mathbf{R}^{el^h} - (\mathbf{M}_c{}^{el^h} - \mathbf{M}_l{}^{el^h})\,\delta\mathbf{U}^h]. \tag{3.16}$$

The calculation is followed by computing the antidiffusive flux array $\mathbf{F}^{el^l}$ from the low-order Taylor–Galerkin solution $\delta\mathbf{U}^l$. The resulting antidiffusive flux array, $\mathbf{F}^{el^l}$, in each element is computed according to $\mathbf{F}^{el^l} = [\mathbf{M}_l{}^{-1}\mathbf{R}^{el^l}]$. When antidiffusive fluxes $\mathbf{F}^{el^h}$ and $\mathbf{F}^{el^l}$ become available, we can calculate the corrected antidiffusive flux array $\mathbf{F}^{el^c}$ [17]. The filtering processes finish with the calculation of $\mathbf{U}^{n+1}$ by means of $\mathbf{U}^{n+1} = \mathbf{U}^l + \mathcal{A}_{el=1}^{n_{el}}(\mathbf{F}^{el^c})$.

**4. Solution solvers.** The iterative solution solver is a strong rival to its direct counterpart because it is less prone to fill-in problems. However, though the storage problem can be considerably resolved, iterative methods have shortcomings of their own. Chief among these shortcomings is the poor control of convergence behavior. Due to space limitations, we will confine our review to iterative solvers based on the minimization concept. The conjugate gradient method of Hestenes and Stiefel [31], considered to be the pioneering work of this class of solvers, works effectively only for a matrix equation having clustered eigenvalues and suffers from pivoting breakdown when matrix symmetry is lost. Refinement of this Krylov subspace method in order to overcome the matrix asymmetry difficulty has been the primary focus of research during the last two decades.

In the literature, nonstationary iterative methods, which have the ability to resolve matrix asymmetry, are frequently referred to [32]. The Chebyshev methods are applicable only to positive definite equations [33]. Also, use of this class of methods requires knowledge of the eigenvalue spectrum a priori. To circumvent deficiencies in irregular convergence behavior and the indispensable transpose operation of the coefficient matrix inherent in the bi-conjugate gradient (Bi-CG) method [34], the Arnoldi or Lanczos algorithms were proposed. Like the Arnoldi algorithm, the GMRES method [19] iterates the approximation solution through use of a self-orthogonal sequence. Due to the prohibitive storage demand, the residual can be minimized optimally by adding a restart capability. In the iteration, no more than $n$ steps are needed for an $n$ by $n$ matrix to reach convergence.

Within the Lanczos framework, product methods, such as conjugate gradient squares (CGS) [35], quasi-minimal residual (QMR) [36], and Bi-CGSTAB [18], are

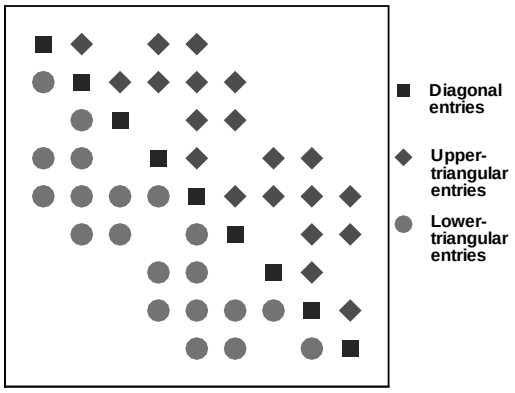



FIG. 4.1. *An illustration of the sparse matrix assembled from* $2 \times 2$ *elements.*

preferable for tackling equation asymmetry. The exploration of a set of dual orthogonal vectors is the building block of this class of methods. The QMR method of Freund and Nachtigal [36] was designed to avoid irregular convergence behavior. This method, unfortunately, suffers from the need to transpose the stiffness matrix. CGS, on the other hand, avoids the need for matrix transpose but inhibits irregular convergence behavior because it accommodates the same contraction polynomial as does Bi-CG. Besides the transpose-free version of QMR [37], the Bi-CGSTAB method of Van der Vorst [18] is a rational alternative. Bi-CGSTAB iterates the approximation solution in terms of unsymmetric Lanczos vectors, in conjunction with the local minimization method GMRES(1). Through manipulation of equal-order contraction polynomials of different kinds, one can dispense with transpose matrix procedures and suppress irregular convergence behavior. Nevertheless, much work still needs to be done so that pivoting breakdown and Lanczos breakdown can be avoided.

The choice of an appropriate solver for obtaining finite element solutions depends on the type of matrix equations employed. Take matrix equations constructed from $2 \times 2$ elements as an example; the sparse matrix, as shown in Figure 4.1, is found to be unsymmetric. Further eigenvalue analysis reveals that the matrix equations are indefinite, thus limiting application of conventional iterative solvers. In this study, we will consider two state-of-the-art iterative solvers and attempt to make a definite assessment of the multifrontal direct solver [30]. Two iterative solvers are the GMRES solver of Saad and Schultz [19] and the Bi-CGSTAB iterative solver of Van der Vorst [18]. These iterative solvers are variants of conjugate gradient methods.

**4.1. The GMRES iterative solver.** We will consider first the GMRES method of Saad and Schultz [19]. This nonstationary iterative solution solver is considered to be an extension of minimal residual (MINRES) which can be used to solve unsymmetric matrix equations. A sequence of tridiagonal matrices is used to obtain a progressively improved distribution of the eigenvalues of the original non-Hermitian linear stiffness matrix system. GMRES iterates the solution with the aid of Arnoldi vectors to overcome the difficulty of the nonsymmetry of the matrix equations. The employed Arnoldi algorithm involves partial tridiagonalization of the original matrix equations using one set of orthogonal vectors $\underline{\mathbf{Q}}$ to yield $\underline{\mathbf{Q}}^T \underline{\underline{\mathbf{A}}}\, \underline{\mathbf{Q}} = \underline{\mathbf{H}}$, where $\underline{\mathbf{H}}$ is

the Hessenberg reduction. The column-by-column generation of $\underline{\mathbf{Q}}$ has the property of $\underline{\mathbf{Q}}^T\,\underline{\mathbf{Q}} = \underline{\mathbf{I}}$ (i.e., identity matrix).

GMRES follows the modified Gram–Schmidt orthogonalization procedure. It invokes a restart capability to control the storage requirement. In GMRES, the main steps are as follows:

Set $\underline{\mathbf{x}}_0$ as an initial guess
**For** $j = 1, 2, \ldots$
  **Solve** $\underline{\mathbf{r}}$ from $\underline{\mathbf{r}} = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}_0$ ⟵ element-by-element procedure
  $\underline{\mathbf{v}}_1 = \underline{\mathbf{r}}/\|\underline{\mathbf{r}}\|_2$
  $s := \|\underline{\mathbf{r}}\|_2$
  **for** $i = 1, 2, 3, \ldots, m$
    **Solve** $\underline{\mathbf{w}}$ from $\underline{\mathbf{w}} = \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{v}}_i$ ⟵ element-by-element procedure
    **for** $k = 1, \ldots, i$
      $h_{k,i} = (\underline{\mathbf{w}}, \underline{\mathbf{v}}_k)$
      $\underline{\mathbf{w}} = \underline{\mathbf{w}} - h_{k,i}\,\underline{\mathbf{v}}_k$
    **end**
    $h_{i+1,i} = \|\underline{\mathbf{w}}\|_2$
    $\underline{\mathbf{v}}_{i+1} = \underline{\mathbf{w}}/h_{i+1,i}$
    apply $J_1, \ldots, J_{i-1}$ on $(h_{1,i}), \ldots, h(i+1, i))$
    construct $J_i$, acting on the $i$th and $(i+1)$st components of $h_{\cdot,i}$,
      such that the $(i+1)$st component of $J_i\,h_{\cdot,i}$ is with the value of 0
    $s := J_i\,s$
    if $s(i+1)$ is small enough, then (**UPDATE** $(\tilde{\underline{\mathbf{x}}}, i)$ and quit )
  **end**
  **UPDATE** $(\tilde{\underline{\mathbf{x}}}, m)$
**End**

The **UPDATE** $(\tilde{\underline{\mathbf{x}}}, i)$ procedure is as follows:

Compute $\underline{\mathbf{y}}$ from $\underline{\underline{\mathbf{H}}}\,\underline{\mathbf{y}} = \underline{\mathbf{s}}$,
  in which the upper $i \times i$ triangular part of $\underline{\underline{\mathbf{H}}}$ has $h_{i,j}$ as its elements,
  $\underline{\mathbf{s}}$ is the first $i$ components of $s$
$\tilde{\underline{\mathbf{x}}} = \underline{\mathbf{x}}_0 + \underline{\mathbf{y}}_1\underline{\mathbf{v}}_1 + \underline{\mathbf{y}}_2\underline{\mathbf{v}}_2 + \cdots + \underline{\mathbf{y}}_i\underline{\mathbf{v}}_i$
$s_{i+1} = \|\underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\tilde{\underline{\mathbf{x}}}\|_2$ ⟵ element-by-element procedure
if $\tilde{\underline{\mathbf{x}}}$ is accurate enough, then quit
else $\underline{\mathbf{x}}_0 = \tilde{\underline{\mathbf{x}}}$.

In the above, the inner product coefficients $\|w^i\|$ and $(w^i, v^k)$ are stored in a Hessenberg matrix. Upon obtaining the values of $y_k$, which are designed to minimize the residual norm $\|\underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}^{(j)}\|$, the GMRES iterations are constructed as $x^j = x^0 + \sum_{i=1} y_i v^i$. GMRES will converge in no more than $n$ iterations when an $n$ by $n$ matrix equation is solved. This is practically infeasible since the storage and computational requirements are prohibitive if $n$ is large. In fact, the crucial factor for successful application of GMRES lies in the restart coded in the program. For this reason, the restarting capability is a built-in feature that can yield the above restarted GMRES($m$), where $m$ denotes the termination number of iteration. The choice of $m$, however, has no theoretical foundation and thus is very difficult to determine. Since there is no definitive rule for the choice of $m$, we determine it through numerical experiments. After conducting extensive investigations, we consider $m = 5$ in all the calculations. For additional details of GMRES($m$), see Van der Vorst [18].

**4.2. The Bi-CGSTAB iterative solver.** The Bi-CG method of Fletcher [34] suffers from instability problem arising from the unsymmetric Lanczos process when

it is used to solve the non-Hermitian system of equations. As a result, considerable effort has been directed toward developing a more stable algorithm. The CGS method is considered as a variant of Bi-CG, known as Bi-CGSTAB. Since the Bi-CGSTAB method is known for its smooth approach to convergence, we will consider this method in our assessment study.

Like the Bi-CG method, Bi-CGSTAB iterates the approximate solution by means of unsymmetric Lanczos vectors. The difficulty arises from the nonsymmetry property of the finite element stiffness matrix $\underline{\underline{\mathbf{A}}}$, which results in the nonexistence of the orthogonal tridiagonalization $\underline{\mathbf{Q}}^T \underline{\underline{\mathbf{A}}}\, \underline{\mathbf{Q}} = \underline{\mathbf{T}}$, where $\underline{\mathbf{T}}$ is a tridiagonal matrix. As a result, partial tridiagonalization of $\underline{\underline{\mathbf{A}}}$ is needed to implement the algorithm. The unsymmetric Lanczos algorithm allows partial tridiagonalization of $\underline{\underline{\mathbf{A}}}$ by making use of two sets of biorthogonal vectors. This approach involves computing columns of $\underline{\mathbf{Q}}$ and $\underline{\mathbf{P}}$, which are subject to $\underline{\mathbf{P}}^T \underline{\mathbf{Q}} = \underline{\mathbf{I}}$, so that $\underline{\mathbf{P}}^T \underline{\underline{\mathbf{A}}}\, \underline{\mathbf{Q}} = \underline{\mathbf{T}}$ is tridiagonal.

It has been known for quite some time that effective use of iterative methods depends highly upon the nonzero profile of the coefficient matrix. The strategies of ordering nodal points and allocating working variables are essential because they have a direct effect on the matrix bandwidth and, thus, matrix sparsity. A means of storing the matrix in the core memory is needed in the finite element analysis, where sparse matrix equations are encountered. Like the compressed matrix used in the finite difference setting, we can store a matrix at the element level so as to dispense with unnecessary storage of voids. This motivates us to conduct finite element analysis on an element-by-element basis. In this paper, we incorporate the element-by-element capability into the Bi-CGSTAB of Van der Vorst [18]:

Compute $\underline{\mathbf{r}}_0 = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\, \underline{\mathbf{x}}_0$ for an initial guess vector $\underline{\mathbf{x}}_0$
Choose $\overline{\underline{\mathbf{r}}}$, such that $(\overline{\underline{\mathbf{r}}}, \underline{\mathbf{r}}_0) \neq 0$
**For** $i = 1, 2, \ldots$
  $\rho_{i-1} = (\overline{\underline{\mathbf{r}}},\ \underline{\mathbf{r}}_{i-1})$
  **if** $\rho_{i-1} < \epsilon_1$ [**near break down**]
  **if** $i = 1$
    $\underline{\mathbf{p}}_i = \underline{\mathbf{r}}_{i-1}$
  **else**
    $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
    $\underline{\mathbf{p}}_i = \underline{\mathbf{r}}_{i-1} + \beta_{i-1}\,(\underline{\mathbf{p}}_{i-1} - \omega_{i-1}\,\underline{\mathbf{v}}_{i-1})$
  **endif**
  $\underline{\mathbf{v}}_i = \sum_{elem} (\underline{\underline{\mathbf{A}}}_{elem}\, \underline{\mathbf{p}}_i)$ ⟵ element-by-element procedure
  $\alpha_i = \rho_{i-1}/(\overline{\underline{\mathbf{r}}},\ \underline{\mathbf{v}}_i)$
  if $(\overline{\underline{\mathbf{r}}},\ \underline{\mathbf{v}}_i) < \epsilon_2$ [**near break down**]
  $\underline{\mathbf{s}} = \underline{\mathbf{r}}_{i-1} - \alpha_i\, \underline{\mathbf{v}}_i$
  $\underline{\mathbf{t}} = \sum_{elem} (\underline{\underline{\mathbf{A}}}_{elem}\, \underline{\mathbf{s}})$ ⟵ element-by-element procedure
  **if** $\|\underline{\mathbf{s}}\|_2 < \epsilon$
    $\omega_i = 0$
  **else**
    $\omega_i = (\underline{\mathbf{t}},\ \underline{\mathbf{s}})/(\underline{\mathbf{t}},\ \underline{\mathbf{t}})$
  **endif**
  $\underline{\mathbf{x}}_i = \underline{\mathbf{x}}_{i-1} + \alpha_i\, \overline{\underline{\mathbf{p}}}_i + \omega_i\, \overline{\underline{\mathbf{s}}}$
  $\underline{\mathbf{r}}_i = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\, \underline{\mathbf{x}}_i$
  check convergence; continue if necessary $(\omega_i \neq 0)$
**End**

For the two iterative methods considered here, the calculation is terminated when the

residual-norm criterion $\|\underline{\mathbf{r}}\|_2 < 10^{-10}$ is satisfied.

**4.3. Multifrontal direct solver.** One of the significant advances in finite element computations was the frontal direct solver developed in 1970 [21]. The frontal solver begins by assembling the matrix for each element. This is followed by incorporating element matrices into the global system of matrices. The elimination of equations is allowed whenever possible, rather than assembling the whole system of elementary finite element matrices. Instead, we examine whether there exists any row which corresponds to the fully contributed nodes; if there is, we store the row, the variables associated with it, and the right-hand side, and then eliminate this row. This process continues until all the elements have been assembled and the elimination procedure is completed. The calculation of solutions is followed by performing backward-substitution.

The multifrontal direct solver can be refined in different ways. The most important solver of this kind is the one developed in 1983 [20]. As the name indicates, many frontal matrices are involved in the course of applying multifrontal solver. The matrix is divided into several balanced substructures. A tree structure is needed to define the order of assembly of element matrices. After the finite element mesh is partitioned, a frontal method is applied to each user's defined substructure to eliminate the interior nodes. A set of substructure matrices is thus generated to complete the elimination process. This is followed by backward-substitution to obtain finite element solutions.

**5. Numerical results.** We will first consider test problems which are amenable to analytical solutions in order to demonstrate the validity and usefulness of the TG-FCT finite element model. The first problem is shown schematically in Figure 5.1. In the square of unit length, a sharp scalar profile was set as the initial condition. The centroid of the square profile was located at (0.25, 0.25). This initially discontinuous scalar profile was transported in the flow specified by $u = \sqrt{2}/2$, $v = \sqrt{2}/2$. Rectangular Cartesian grids were uniformly overlaid on the region of interest. The grid spacings were set as $\Delta x = \Delta y = 0.01$, and the time increment for this study was chosen to be $\Delta t = 0.001$. The calculation for this study was terminated at $t = 0.5$. The numerical models were run using iterative and direct solvers with a tolerance equal to $10^{-10}$. The result shown in Figure 5.2 clearly indicates that the passive scalar was well predicted without observable oscillations. This validation test shows that the scheme adopted here has the ability to resolve discontinuities.

We now turn to making a comparison of the employed frontal, GMRES(5), and Bi-CGSTAB solution solvers. The user, system, and CPU times shown in Table 5.1 are obtained for the codes run on an Intel Celeron/466 MHz processor. As this table shows, GMRES(5) consumes only 1/9 CPU time as that computed by the frontal solver. As for the Bi-CGSTAB solver, it is slower than the GMRES(5) by a factor of 7/9. One plausible reason for explaining the savings in CPU time of the GMRES solver is due to its relatively regular convergence. Taking $t = \Delta t$ as an example, the convergent histories, shown in Figure 5.3, clearly show that the specified tolerance is reached much more quickly when GMRES is employed. As for the CPU time spent in this arbitrarily chosen time step, the ratio of $\text{CPU}|_{\text{GMRES(5)}}$ / $\text{CPU}|_{\text{Bi-CGSTAB}}$ is equal to 4/5.

The next problem was intended to test the ability of the scheme to resolve discontinuous field variables in shallow water equations. In this validation, we solve for a one-dimensional dam-break problem using the proposed two-dimensional finite element code. The problem, shown in Figure 5.4, involved a dam 100 m in length.

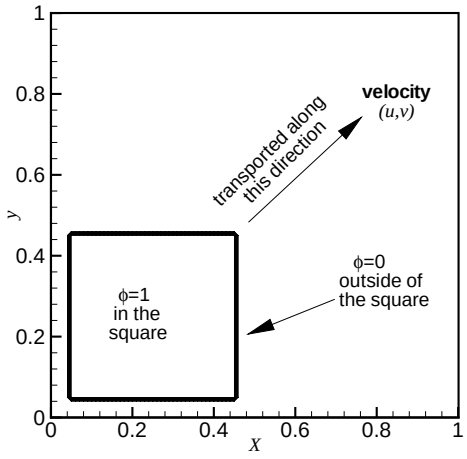
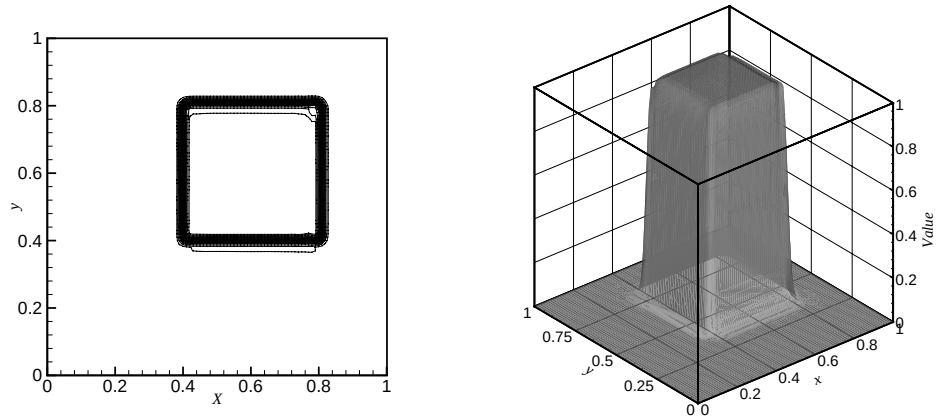



FIG. 5.1. *The schematic illustration of the scalar transport problem.*




(a) (b)

FIG. 5.2. (a) *The contours of the computed solution at* $t = 0.5$*;* (b) *the three-dimensional view of the solution at* $t = 0.5$*. The grid size used for this study is* $\Delta x = \Delta y = 10^{-2}$.

TABLE 5.1

*The comparison of CPU times (in seconds) for solving the problem, schematically shown in Figure* 5.1*, using the frontal, GMRES, and Bi-CGSTAB solvers.*

| Method | User time | System time | CPU time |
|---|---|---|---|
| frontal [21] | 20842 | 1187 | 22029 |
| GMRES(5) [19] | 1898 | 664 | 2562 |
| Bi-CGSTAB [18] | 2409 | 664 | 3073 |

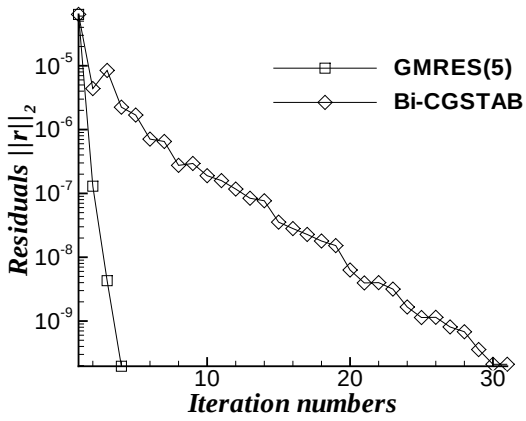



FIG. 5.3. *The convergent histories of the GMRES*(5) *and Bi-CGSTAB methods within a time step* $0 \le t \le \Delta t$ *for the transport problem given in Figure* 5.1.

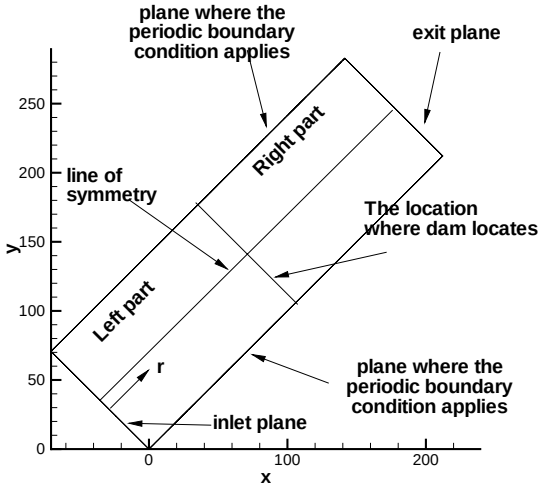



FIG. 5.4. *The configuration of the one-dimensional dam-break problem.*

For the present study, a channel 300 m long and 100 m wide was used and was discretized into 301×7 grids for numerical calculation. The case under investigation was a subcritical flow with a water-height ratio of 2. Both the upstream and downstream boundary conditions remained unchanged during the calculation. For this purpose, the test was run at $t = 10$ s.

Figure 5.5 shows the water height, which compares very favorably with the following analytic data of Stoker [38]:

$$h(\tilde{x},t) = \begin{cases} h_1 & \text{if } \frac{\tilde{x}}{t} \le -\sqrt{gh_1}, \\ (\frac{1}{9g})\left[2\sqrt{gh_1} - \frac{\tilde{x}}{t}\right]^2 & \text{if } -\sqrt{gh_1} \le \frac{\tilde{x}}{t} \le \left[u_m - \sqrt{gh_m}\right], \\ h_m & \text{if } \left[u_m - \sqrt{gh_m}\right] \le \frac{\tilde{x}}{t} \le s, \\ h_2 & \text{if } s \le \frac{\tilde{x}}{t} \le \infty. \end{cases} \tag{5.1}$$

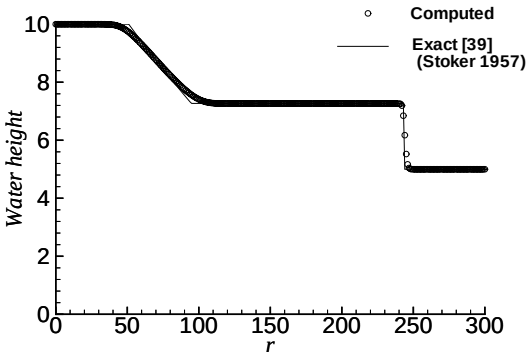



FIG. 5.5. *The comparison of the solution of the one-dimensional dam-break problem with the Stoker's analytic solution. The grid size for this problem is* $\Delta r = 1$.

We denote here $\tilde{x} = x - x_0$, where $x_0$ is the location of the discontinuity. In the above, $h_m$ and $u_m$ are the water height and velocity in the middle of this channel, and their values are related to the shock propagation speed $s$:

$$h_m = \frac{1}{2}\left[\sqrt{1 + \frac{8s^2}{gh_2}} - 1\right] h_2, \tag{5.2}$$

$$u_m = s - \frac{gh_2}{4s}\left[1 + \sqrt{1 + \frac{8s^2}{gh_2}}\right]. \tag{5.3}$$

The shock speed $s$ is the positive real root of the following equation:

$$u_m + 2\sqrt{gh_m} - 2\sqrt{gh_1} = 0. \tag{5.4}$$

It is seen from the computed solutions that the shock wave can be resolved within 4 mesh points. No postshock oscillations are observed in the solution. Also, the improved accuracy is attributable to the high-order Taylor–Galerkin scheme, which is applied in regions away from the discontinuity.

We will now consider wave propagation in a basin of simple geometry. Figure 5.6 shows a typical configuration extensively used to study shallow water where bore waves may develop. At the midpoint of the square basin, a dam with a width of 10 m equally divides the water into two parts. On both sides of this idealized dam, the water elevations have a height ratio other than 1. At time $t = 0^+$, the dam is partially broken, leading to a breach with a width of 75 m. The resulting flow pattern in this partially breached dam depends on whether it is classified as being subcritical or supercritical. The case examined here is a subcritical flow with $h_L/h_R = 2$, where $h_L\,(\equiv 10$ m) denotes the initial water elevation on the left side of the basin while $h_R\,(\equiv 5$ m) represents the water height on the right side. Given that the ratio $h_L/h_R$ has a value other than 1, water proceeds toward the downstream side through the breach, which is located in the region $y = 95$ m to $y = 170$ m. When the dam breaks, a bore wave starts to propagate forward and spread laterally. At the same time, a negative depression wave spreads upstream. In addition, a standing wave will appear

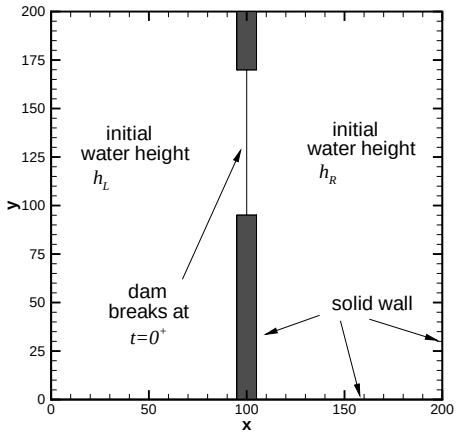



FIG. 5.6. *The configuration of the two-dimensional dam-break problem.*

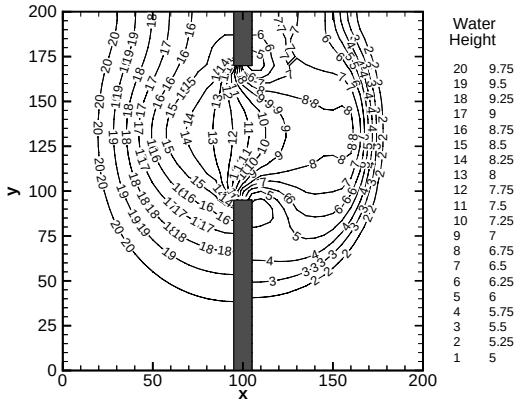



FIG. 5.7. *The contours of water height of the two-dimensional dam-break problem.*

due to the reduction of velocity at the two side walls. Our goal is to numerically predict the time-evolving propagation and spreading of the wave into the reservoir.

With the Courant number set as 0.1, the numerical code was run on a domain of uniform spacing, $\Delta x = \Delta y = 5$. Figure 5.7 plots the contour values of the water elevation obtained using the proposed TG-FCT method at $t = 7.2$ s. These values show that the right traveling bore wave and left traveling depression wave have both been predicted. The results also reveal the ability of the FCT scheme to capture sharp solutions, as seen from the abrupt depression of the water surface elevation in the vicinity of the breach edge. The appearance of this sharp depression wave is theoretically justified since strong rarefactions are established in the inviscid flow regime where large velocity gradients appear. In light of this fact, we plot in Figure 5.8 the velocity vector to show the sharp depression in the water surface elevation around the breach edge. From this velocity vector plot, we are led to conclude that the regions with large velocity gradients observed near the edge of the breach appear to be those which are most subject to abrupt depression in the water surface elevation.

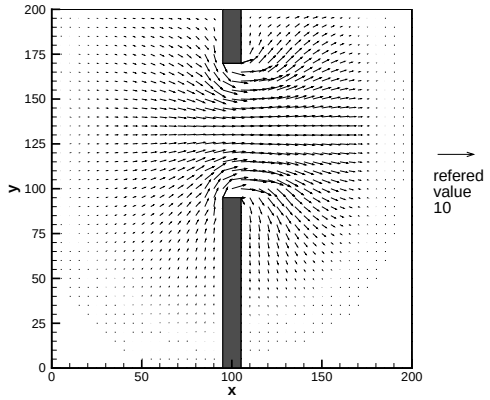



FIG. 5.8. *The velocity vector of the two-dimensional dam-break problem.*

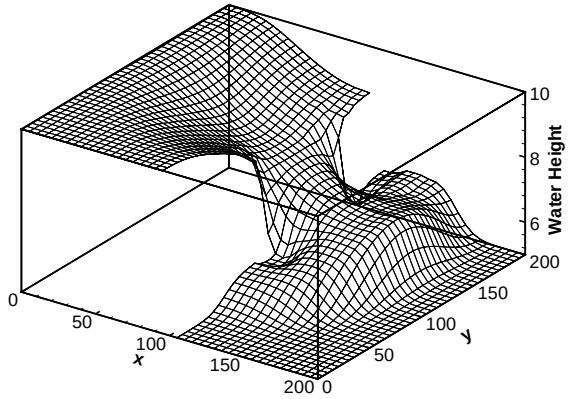



FIG. 5.9. *The three-dimensional view of the water height of the two-dimensional dam-break problem.*

To present the solution clearly, we plot in Figure 5.9 a three-dimensional water surface at $t = 7.2$ s after the dam breaks.

Having demonstrated the advantage of using the GMRES(5) solver over the Bi-CGSTAB solver, we simply consider the GMRES solver in the shallow water calculation and make a computational assessment with the two employed direct solvers. Table 5.2 shows that GMRES(5) is ten times faster than the frontal solver. As for the multifrontal direct solver, it takes only 1/3 of the CPU time needed for the direct solver. Note that the times summarized in Table 5.2 are obtained on an SGI Origin 2000 computer. This performance test demonstrates the effective utility of the multifrontal solver and, more importantly, ensures the advantage of applying the GMRES iterative solver to shallow water analyses.

To show that this method is applicable to predicting a more severely changing solution profile, we considered a supercritical flow. The initial water height ratio was prescribed as $h_R/h_L = 0.05$. As Figure 5.10 shows, water heights were captured in a sharp and nonoscillatory way. This test also sheds light on the effectiveness of adding entropy flux to the region where needed (near the sonic point) since no expansion

TABLE 5.2

*The comparison of CPU times (in seconds) for solving the dam-break problem using the frontal, multifrontal, and GMRES solvers.*

| Method | User time | System time | CPU time |
|---|---|---|---|
| frontal [21] | 28961 | 199 | 29160 |
| multifrontal [20] | 9657 | 142 | 9799 |
| GMRES(5) [19] | 2816 | 143 | 2959 |

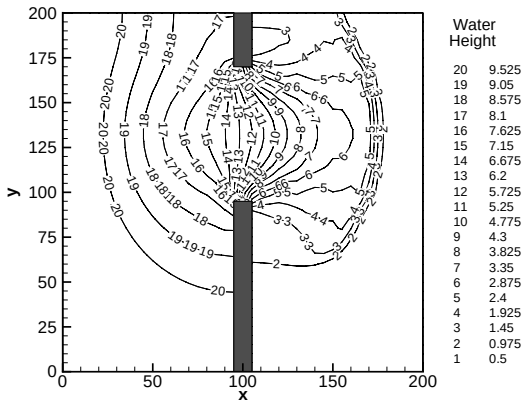



FIG. 5.10. *The contours of water height of the two-dimensional dam-break problem with* $h_R/h_L = 0.05$.

shock is observed. It is thus concluded that the FCT technique incorporated into the Taylor–Galerkin formulation has the ability to suppress dispersive errors near a discontinuity without adding dissipation error which could deteriorate the solution.

**6. Concluding remarks.** We have applied in this paper a generalized Taylor–Galerkin finite element model to simulate shallow water equations in two dimensions. By prescribing different sets of free parameters a priori, we can render the technique suitable for hydraulic problems having sharply or smoothly varying solution profiles. We have applied the FCT filtering scheme to obtain high-resolution solutions. The main idea of developing the high-order TG-FEM model is the adoption of modified equation analysis. As for the free parameters used in the low-order Taylor–Galerkin model, we employ the discrete maximum principle to construct a stiffness matrix of the M-matrix type. The avoidance of numerical oscillations near the discontinuity and the higher level of prediction accuracy in the smooth region make this scheme a robust tool for solving differential equations governing shallow water height. The code was run on several test problems to study the method's performance and the solver's efficiency, with particular attention paid to the shock-capturing ability and the computational advantage. For the purpose of validation, we have chosen ones for which exact solutions are available. These include the scalar equation and the shallow water equations. Numerical results show that field variables were captured in a sharp and nonoscillatory way in both subcritical and supercritical situations. Through computational exercises, we advocate the use of iterative solution solvers. Of two investigated iterative solvers, the GMRES outperforms the Bi-CGSTAB solver. The present study also shows the advantage of applying the multifrontal direct solver over the frontal direct solver as far as the present shallow water analysis is considered.

**Acknowledgment.** The authors would like to thank the National Center for High-Performance Computing (NCHC) for providing the SGI Origin 2000 computer which made this study possible.

REFERENCES

[1] E. F. Toro, *Riemann problems and the WAF method for solving the two-dimensional shallow water equations*, Philos. Trans. Roy. Soc. London Ser. A, 338 (1992), pp. 43–68.
[2] S. K. Godunov, *Finite-difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, Mat. Sb., 47 (1959), pp. 271–306 (in Russian).
[3] B. Van Leer, *Towards the ultimate conservative difference scheme,* II. *Monotonicity and second order combined in a second order scheme*, J. Comput. Phys., 14 (1973), pp. 361–376.
[4] A. Harten, *High-resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.
[5] P. Colella and P. R. Woodward, *The piecewise parabolic method (PPM) for gas dynamics simulation*, J. Comput. Phys., 54 (1984), pp. 174–201.
[6] A. Harten and S. Osher, *Uniformly high-order accurate nonoscillatory schemes.* I, SIAM J. Numer. Anal., 24 (1987), pp. 279–309.
[7] P. Garcia-Navaroo, M. E. Hubband, and A. Priestley, *Genuinely multidimensional upwinding for the* 2D *shallow water equations*, J. Comput. Phys., 121 (1995), pp. 79–93.
[8] H. Paillere, G. Dergez, and H. Deconinck, *Multidimensional upwind schemes for the shallow water equations*, Internat. J. Numer. Methods Fluids, 26 (1998), pp. 987–1000.
[9] T. Molls and F. Molls, *Space-time conservation method applied to Saint Venant equations*, J. Hydr. Engrg., 124 (1998), pp. 501–508.
[10] C. G. Mingham and D. M. Causon, *High-resolution finite-volume method for shallow water flows*, J. Hydr. Engrg., 124 (1998), pp. 605–614.
[11] R. J. Fernema and M. H. Chaudhry, *Explicit methods for* 2*-D transient free-surface flows*, J. Hydr. Engrg., 116 (1990), pp. 1013–1034.
[12] L. Fraccarollo and E. Toro, *Experimental and numerical assessment of the shallow water model for two dimensional dam-break type problems*, J. Hydr. Res., 33 (1995), pp. 843–864.
[13] R. Garcia and R. Kahawita, *Numerical solution of the St. Venant equations with MacCormack finite-difference scheme*, Internat. J. Numer. Methods Fluids, 6 (1986), pp. 507–527.
[14] D. H. Zhao, H. W. Shen, J. S. Lai, and G. Q. Tabios, *Approximate Riemann solvers in FVM for* 2D *hydraulic wave modeling*, J. Hydr. Engrg., 122 (1996), pp. 692–702.
[15] N. D. Katopodes and T. Strelkoff, *Computing two-dimensional dam-break flood waves*, J. Hydr. Div., 110 (1988), pp. 1269–1288.
[16] J. P. Boris and D. L. Book, *Flux corrected transport, SHASTA, a fluid transport algorithm that works*, J. Comput. Phys., 11 (1973), pp. 38–69.
[17] S. T. Zalesak, *Fully multidimensional flux-corrected transport algorithm for fluids*, J. Comput. Phys., 31 (1979), pp. 335–362.
[18] H. A. Van der Vorst, *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear system*, SIAM J. Sci. Statist. Comput. 13 (1992), pp. 631–644.
[19] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear system*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[20] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.
[21] B. M. Irons, *A frontal solution program for finite element analysis*, Internat. J. Numer. Methods Engrg., 4 (1970), pp. 5–32.
[22] J. Donea, *A Taylor-Galerkin method for convective transport problems*, Internat. J. Numer. Methods Engrg., 20 (1984), pp. 101–119.
[23] W. H. Sheu and C. C. Fang, *A numerical study of nonlinear propagation of disturbances in two-dimensions*, J. Comput. Acoustics, 4 (1996), pp. 291–319.
[24] M. H. Chaudhry, *Open-Channel Flow*, Prentice–Hall, Englewood Cliffs, NJ, 1993.
[25] G. A. Sod, *Numerical Methods in Fluid Dynamics*, Cambridge University Press, Cambridge, UK, 1985.
[26] T. Meis and U. Marcowitz, *Numerical Solution of Partial Differential Equations*, Appl. Math. Sci. 32, Springer-Verlag, New York, 1981.
[27] T. Ikeda, *Maximal Principle in Finite Element Models for Convection-Diffusion Phenomena*, Lecture Notes Numer. Appl. Anal. 4, Kinokuniya, Tokyo, Japan, 1983.

[28] M. Ahués and M. Telias, *Petrov-Galerkin Scheme for the Steady State Convection Diffusion Equation*, Finite Elements in Water Resources 2/3, 1982.
[29] A. Harten, P. D. Lax, and B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Rev., 25 (1983), pp. 35–61.
[30] Y. E. Campbell, *Multifrontal Algorithms for Sparse Inverse Subsets and Incomplete LU Factorization*, Technical Report TR-95-025, Computer and Information Sciences Department, University of Florida, Gainesville, FL, 1995.
[31] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., North Oxford Academic, Oxford, 1986.
[33] R. Varga, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.
[34] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems*, Lecture Notes in Math. 506, Springer-Verlag, New York, 1976, pp. 73–89.
[35] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.
[36] R. Freund and N. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear system*, Numer. Math., 60 (1991), pp. 315–339.
[37] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.
[38] J. J. Stoker, *Water Waves*, Interscience, New York, 1957.