Multi-grid Beam and Warming scheme for the simulation of unsteady flow in an open channel

Majid Rahimpour[1]* and Ali Tavakoli[2]
[1] *Water Engineering Department, Shahid Bahonar University of Kerman, Iran*
[2] *Department of Mathematics, Vali-e-Asr University of Rafsanjan, Iran*

Abstract

In this paper, a multi-grid algorithm is applied to a large-scale block matrix that is produced from a Beam and Warming scheme. The Beam and Warming scheme is used in the simulation of unsteady flow in an open channel. The Gauss-Seidel block-wise iteration method is used for a smoothing process with a few iterations. It is also shown that the governing equations determine the type of prolongation and restriction operators for the multi-grid algorithm.

Keywords: Multi-grid algorithm, Beam and Warming scheme, open channel flow

Introduction

Unsteady flow is of great interest to hydraulic engineers. Such flows can be described by the Saint-Venant equations which consist of equations for conservation of mass and momentum. The Saint-Venant equations are also non-linear hyperbolic partial differential equations. However, a general closed-form solution of these equations is not available, except for certain special simplified conditions, and they must be solved using an appropriate numerical technique. Among different numerical methods, the implicit finite-difference method and finite element method have been widely used for the solution of 1-dimensional unsteady open-channel flow problems (Aureli et al., 2008; Catellal, 2008; Choi and Molinas, 1993; Jha et al., 1994; Nguyen and Kawano, 1995; Sen and Garg, 2002; Tseng and Chu, 2000; Venutelli, 2002).

The discretisation of partial differential equations leads to very large systems of equations. For 2-dimensional problems, several 10 thousands of unknowns are not usual, and in 3 spatial dimensions more than 1 million unknowns can be reached very easily. Therefore, iterative methods like Jacobi or Gauss-Seidel relaxations have been used. Nevertheless, the main defect of iterative methods is that these will work very well in a few iterations, but after that these methods will be converged slowly. Multi-grid algorithms will solve this problem (Bramble and Pasciak, 1987; Tavakoli and Kerayechian, 2007; Tavakoli, 2010). Multi-grid methods are the fastest known methods for the solution of the large systems of equations arising from the discretisation of partial differential equations (for more details see Bramble, 1993).

Governing equations

The governing equations based on the conservation of mass and momentum for 1-dimensional unsteady open-channel flow in a prismatic channel of arbitrary cross-section can be expressed as (see Abbott, 1979; Chaudhry, 1993; Chow, 1959 for details):

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\mathbf{S}=0 \qquad or \qquad \mathbf{U}_t+\mathbf{F}_x+\mathbf{S}=0 \tag{1}$$

where:

t represents time
x represents longitudinal distance

$$\mathbf{U}=\begin{bmatrix} A \\ uA \end{bmatrix}; \qquad \mathbf{F}=\begin{bmatrix} uA \\ u^2A+gF_h \end{bmatrix};$$

$$\mathbf{S}=\begin{bmatrix} 0 \\ -gA(S_0-S_f) \end{bmatrix} \tag{2}$$

where:

A is the cross-sectional area of flow
u the velocity, g the gravitation acceleration
F_h the hydrostatic pressure force term
S_f the friction slope, and S_0 the bed slope

The friction slope is computed by the Manning formula:

$$S_f=\frac{n_c^2 u|u|}{R^{4/3}} \tag{3}$$

where:

n_c is the Manning coefficient
R the hydraulic radius.

The hydrostatic pressure force term may be expressed as:

$$F_h=\int_0^h (h-\eta)W(\eta)d\eta \tag{4}$$

where:

h is the flow depth
η the integration variable indicating distance from channel bottom
$W(\eta)$ the water-surface width at distance η from the channel bottom

The hydrostatic pressure force term for rectangular and trapezoidal channels becomes:

$$F_h=\frac{1}{2}b_w h^2+\frac{zh^3}{3} \tag{5}$$

* To whom all correspondence should be addressed.
☎ +98 913 199 0614; fax: +98 341 3220733;
e-mail: rahimpour@mail.uk.ac.ir
Received 8 March 2010; accepted in revised form 7 March 2011.

where:

b_w is the base width of channel

z the $_s$ide slope of the channel.

The governing equations are based on the assumption of hydrostatic-pressure distribution, incompressibility of water, sufficiently small bottom slope of the channel, and negligible wind stress and Coriolis force. Eq. (1) can be expressed in quasi-linear form as:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{M}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{S} = 0 \tag{6}$$

where:

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -u^2 + gD & 2u \end{bmatrix} \tag{7}$$

is the Jacobian matrix and

$D = A/W$ (η) is the hydraulic depth.

Furthermore, since the matrix **M** has independent and real eigen-vectors, it can be written in diagonalised form as:

$$\mathbf{M} = \frac{1}{2c}\begin{bmatrix} 1 & -1 \\ u+c & -(u-c) \end{bmatrix}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}\begin{bmatrix} -(u-c) & 1 \\ -(u+c) & 1 \end{bmatrix} \tag{8}$$

where:

c is the wave celerity expressed as $\sqrt{gD}$, and

the λ_i values are eigen-values of **M** giving the characteristic directions.

The eigen-values are given by:

$$\lambda_1 = u + c; \quad and \quad \lambda_2 = u - c \tag{9}$$

The matrix **M** can now be split into 2 components, positive and negative, by the testing sign of the eigen-values. This may be done as follows:

$$\mathbf{M} = \mathbf{M}^+ + \mathbf{M}^- \tag{10}$$

where:

$$\lambda_i^+ = \max(\lambda_i, 0); \quad and \quad \lambda_i^- = \lambda_i - \lambda_i^+$$

Beam and Warming scheme

The finite-difference approximation for flow variables **U** at the higher time level, i.e.: $\mathbf{U}^{k+1}$, can be written as (Beam and Warming, 1976; Chaudhry, 1993; Jha et al., 1994):

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \frac{\Delta t}{2}[(\frac{\partial \mathbf{U}}{\partial t})^{k+1} + (\frac{\partial \mathbf{U}}{\partial t})^k] \tag{11}$$

By substituting the value of $\mathbf{U}_t$ from Eq. (1) into Eq. (11):

$$\mathbf{U}^{k+1} = \mathbf{U}^k - \frac{\Delta t}{2}[(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^{k+1} + (\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^k] \tag{12}$$

The terms $\mathbf{F}^{k+1}$ and $\mathbf{S}^{k+1}$ in Eq. (12) are nonlinear and may be linearised as follows. The Taylor series expansion of $\mathbf{F}^{k+1}$ may be written as:

$$\begin{aligned} \mathbf{F}^{k+1} &= \mathbf{F}^k + \Delta t(\frac{\partial \mathbf{F}}{\partial t})^k + O(\Delta t)^2 \\ &= \mathbf{F}^k + \Delta t(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial t})^k + O(\Delta t)^2 \\ &= \mathbf{F}^k + \mathbf{M}(\frac{\partial \mathbf{U}}{\partial t})^k \Delta t \\ &= \mathbf{F}^k + \mathbf{M}(\mathbf{U}^{k+1} - \mathbf{U}^k) \end{aligned} \tag{13}$$

Similarly:

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \mathbf{B}(\mathbf{U}^{k+1} - \mathbf{U}^k) \tag{14}$$

where:

M and **B** are the Jacobians of **F** and **S** with respect to **U**, respectively.

The matrix **M** in Eq. (7) is replaced in the Eq. (13) and **B** is given as:

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ -gS_0 - \dfrac{4gn_c^2 u|u|}{R^{4/3}} & \dfrac{gn_c^2|u|}{R^{4/3}} \end{pmatrix}. \tag{15}$$

Substituting Eqs. (13) and (14) into Eq. (12) and simplifying

$$\begin{aligned} \mathbf{U}^{k+1} - \mathbf{U}^k &= -\frac{\Delta t}{2}[(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^{k+1} + (\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^k] \\ &= -\frac{\Delta t}{2}(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^{k+1} + (-\Delta t + \frac{\Delta t}{2})(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^k \\ &= -\frac{\Delta t}{2}[(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^{k+1} - (\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^k] - \Delta t(\frac{\partial \mathbf{F}}{\partial x} + \mathbf{S})^k \\ &= -\frac{\Delta t}{2}[\frac{\partial(\mathbf{F}^k + \mathbf{M}^k(\mathbf{U}^{k+1} - \mathbf{U}^k))}{\partial x} + \mathbf{S}^{k+1} - \frac{\partial \mathbf{F}^k}{\partial x} - \mathbf{S}^k] - \Delta t(\mathbf{M}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{S})^k \\ &= -\frac{\Delta t}{2}[\frac{\partial(\mathbf{M}^k\mathbf{U}^{k+1})}{\partial x} - \frac{\partial(\mathbf{M}^k\mathbf{U}^k)}{\partial x} + \mathbf{B}^k(\mathbf{U}^{k+1} - \mathbf{U}^k)] - \Delta t(\mathbf{M}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{S})^k \end{aligned} \tag{16}$$

Therefore:

$$\begin{aligned} &\mathbf{U}^{k+1} + \frac{\Delta t}{2}[\frac{\partial(\mathbf{M}^k\mathbf{U}^{k+1})}{\partial x} + \mathbf{B}^k\mathbf{U}^{k+1}] \\ &= \mathbf{U}^k + \frac{\Delta t}{2}[\frac{\partial(\mathbf{M}^k\mathbf{U}^k)}{\partial x} + \mathbf{B}^k\mathbf{U}^k] - \Delta t(\mathbf{M}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{S})^k \end{aligned} \tag{17}$$

Introducing the split form of matrix **M**, as given by Eq. (10), Eq. (17) is written as

$$\begin{aligned} &[\mathbf{I} + \frac{\Delta t}{2}(\frac{\partial \mathbf{M}^+}{\partial x} + \frac{\partial \mathbf{M}^-}{\partial x} + \mathbf{B}\ )^k]\mathbf{U}^{k+1} \\ &= [\mathbf{I} + \frac{\Delta t}{2}[(\frac{\partial \mathbf{M}^+}{\partial x} + \frac{\partial \mathbf{M}^-}{\partial x} + \mathbf{B}\ )^k]\mathbf{U}^k - \Delta t[(\mathbf{M}^+ + \mathbf{M}^-)\frac{\partial \mathbf{U}}{\partial x} + \mathbf{S}]^k \end{aligned} \tag{18}$$

where:

I = α (2 x 2) unit matrix.

The expressions in the parentheses before $\mathbf{U}^{k+1}$ and $\mathbf{U}^k$ are operators on $\mathbf{U}^{k+1}$ and $\mathbf{U}^k$ respectively, not simple products. The space derivatives associated with positive and negative components of **M** are approximated by backward and forward space differences, respectively. Therefore:

$$\frac{\partial(\mathbf{M}^+\mathbf{U})}{\partial x} = \frac{\mathbf{M}_i^+\mathbf{U}_i - \mathbf{M}_{i-1}^+\mathbf{U}_{i-1}}{\Delta x} \tag{19}$$

$$\frac{\partial(\mathbf{M}^-\mathbf{U})}{\partial x} = \frac{\mathbf{M}_{i+1}^-\mathbf{U}_{i+1} - \mathbf{M}_i^-\mathbf{U}_i}{\Delta x} \tag{20}$$

Multi-grid algorithms

Multi-grid algorithms have become a common approach for solving many types of linear problems of the form $Gx = b$. In order to describe the multi-grid algorithm, let Ω be the domain of problem (1). To approximate $\mathbf{U} = \begin{bmatrix} A \\ uA \end{bmatrix}$, we consider a sequence of subintervals Ω^j of Ω determined by a regular subdivision; namely, the set Ω is divided into 2^j equi-distant subintervals at level j = 1,2, ..., J. In addition, we denote G_j as the generated matrix by Beam-Warming scheme at the level j and at the finest

Available on website http://www.wrc.org.za
ISSN 0378-4738 (Print) = Water SA Vol. 37 No. 2 April 2011
ISSN 1816-7950 (On-line) = Water SA Vol. 37 No. 2 April 2011

level $j = J$ we define $G_j = G$. Moreover, we consider $I_{j-1}^{j} : \Omega^{j-1} \to \Omega^{j}$ as an operator from the coarse grid Ω^{j-1} to the fine grid Ω^{j}. Furthermore, we define R_j as a smoothing operator for level j. For example, in the Jacobi relaxation scheme, R_j is defined D_j^{-1} where D_j is the diagonal matrix of G. We denote an approximation of the exact solution x by v and the error, $e = x - v$ by e. Defining the residual by $r = b - Gv$, we observe the critical relationship known as the residual equation, namely $Ge = r$. In brief, there is a recursive application of the 2-grid process. First, an iterative method such as the Jacobi or Richardson relaxation is applied to the fine-grid problem. These iterations have the property that after relaxation the error will be smooth. This in turn means that the error can be accurately represented on a coarse grid. Since the coarse grid is much smaller than the fine grid, it is much less expensive to work on the coarse grid. These facts permit the second part of the process, known as the coarse-grid correction. The fine-grid residual r^j at the level j is computed and restricted to the coarse grid $r^{j-1} = I_j^{j-1} r^j$ at the level $j-1$, where it is used as the right-hand side of the coarse-grid residual equation $G_{j-1} e^{j-1} = r^{j-1}$. This equation is solved and the error thus determined is then interpolated back to the fine grid, where it is used to correct the fine-grid approximation, $v^j \leftarrow v^j + I_{j-1} e^{j-1}$. By recursively solving the coarse-grid equation with this 2-grid process, a multi-grid algorithm is defined. Now we are ready to state the multi-grid algorithm (Bramble and Pasciak, 1987; Bramble et al., 1991).

Multi-grid algorithm

$\mathrm{MG}(\mathrm{j}, \mathrm{x}^0, \mathrm{b_j})$ is the approximate solution of the equation

$$G_j x = \mathrm{bj}$$

Obtained from the j^{th} level iteration with initial guess x^0. At the finest level J, set $b_j := b$ and $G_j := G$

For $j = 0$, $\mathrm{MG}(0, \mathrm{x}^0, \mathrm{b}_0)$ is the solution obtained from a direct method. In other words,

$$\mathrm{MG}(0, \mathrm{x}^0, \mathrm{b}_0) = \mathrm{G}_0^{-1}\mathrm{b}_0.$$

We note that b_0 is determined recursively by Step (2) in the following error correction step. For $j \geq 1$, $\mathrm{MG}(\mathrm{j}, \mathrm{x}^0, \mathrm{b_j})$ is obtained recursively in 7 steps.

Pre-smoothing step:

1. Define x^l for $l = 1, \ldots, m$ by
$$x^l = x^{l-1} + R_j(\mathrm{b_j} - G_j x^{l-1}).$$

Error correction step:

2. Set $\mathrm{b_{j-1}} = I_j^{j-1}(\mathrm{b_j} - G_j x^m)$. (Transformation of residual from fine grid j to coarse grid $j-1$.)
3. Set $q^0 = 0$.
4. (Solving the residual equation) For $i = 1, \ldots, p$ defines error corrector
$$\mathrm{q^i} = \mathrm{MG}(\mathrm{j}-1, \mathrm{q^{i-1}}, \mathrm{b_{j-1}}).$$
If the residual equation in Step (4) is not solvable directly, set $j := j-1$ and go to Step (2). Otherwise:
5. Define $x^{m+1} = x^m + I_{j-1}^{j} q^p$

Post-smoothing step:

6. Define x^l for $l = m+1, \ldots, 2m$ by
$$x^{l+1} = x^l + R_j(b_j - G_j x^l).$$
If $\mathrm{j} \neq \mathrm{J}$, set $m := 2m+1$ and go to Step (5). Otherwise:
7. Define $\mathrm{MG}(\mathrm{J}, \mathrm{x}^0, \mathrm{b}) = \mathrm{x}^{2\mathrm{m}}$.

In this algorithm, m is a positive integer which may vary from level to level and determines the number of pre- and post-smoothing iterations. If $p = 1$, we have a V-cycle multi-grid algorithm. If $p = 2$, we have a W-cycle algorithm. A variable V-cycle algorithm is one in which the number of smoothing m increases exponentially as j decreases (i.e., $p = 1$ and $m = 2^{J-j}$). The above multi-grid is called a symmetric multi-grid algorithm. If the post-smoothing step is removed, it is called a non-symmetric multi-grid algorithm.

Block multi-grid algorithm for Beam-Warming scheme

The differential equation can aid in forming a prolongation operator (Trottenberg et al., 2001). In this section, we use the Beam-Warming scheme to form the interpolation (prolongation) operator. To this end, let $I_{j-1}^{j} : \Omega^{j-1} \to \Omega^{j}$ be a mapping from the coarse grid Ω^{j-1} to the fine grid Ω^{j}. We also assume that:

$$v^j = \left(\begin{bmatrix} v_{0,1}^j \\ v_{0,2}^j \end{bmatrix}, \begin{bmatrix} v_{1,1}^j \\ v_{1,2}^j \end{bmatrix}, \ldots, \begin{bmatrix} v_{n,1}^j \\ v_{n,2}^j \end{bmatrix}\right)$$

and

$$v^{j-1} = \left(\begin{bmatrix} v_{0,1}^{j-1} \\ v_{0,2}^{j-1} \end{bmatrix}, \begin{bmatrix} v_{1,1}^{j-1} \\ v_{1,2}^{j-1} \end{bmatrix}, \ldots, \begin{bmatrix} v_{\frac{n}{2},1}^{j-1} \\ v_{\frac{n}{2},2}^{j-1} \end{bmatrix}\right)$$

are defined on Ω^j and Ω^{j-1}, respectively. To construct the prolongation operator, the values of coarse-grid points will be transformed unchanged, i.e.:

$$\begin{bmatrix} v_{2i,1}^j \\ v_{2i,2}^j \end{bmatrix} = \begin{bmatrix} v_{i,1}^{j-1} \\ v_{i,2}^{j-1} \end{bmatrix} \tag{21}$$

for $0 \leq i \leq \frac{n}{2}$. To construct a non-standard mapping on the middle points, one can use the Beam-Warming scheme. For this purpose, we first denote w^k as the value of w at time t_k:

$$\begin{aligned} \begin{bmatrix} v_{2i+1,1}^j \\ v_{2i+1,2}^j \end{bmatrix} + \frac{\Delta t}{2}\Bigg(\frac{M_{2i+1}^{+} - M_{2i+1}^{-}}{\Delta x} \\ + B_{2i+1}\Bigg)\begin{bmatrix} v_{2i+1,1}^j \\ v_{2i+1,2}^j \end{bmatrix} \\ = \frac{\Delta t}{2\Delta x}\Bigg(M_{2i+2}^{-}\begin{bmatrix} v_{2i+2,1}^j \\ v_{2i+2,2}^j \end{bmatrix} \\ - M_{2i}^{+}\begin{bmatrix} v_{2i,1}^j \\ v_{2i,2}^j \end{bmatrix}\Bigg) \end{aligned} \tag{22}$$

Hence, by Eqs. (21) and (22) we have:

$$\begin{aligned} \begin{bmatrix} v_{2i+1,1}^j \\ v_{2i+1,2}^j \end{bmatrix} = \frac{\Delta t}{2\Delta x}\Bigg[(I + \frac{\Delta t}{2}\left(\frac{M_{2i+1}^{+} - M_{2i+1}^{-}}{\Delta x}\right) \\ + B_{2i+1})^k\Bigg]^{-1}\Bigg((M_{2i+2}^{-})^k\begin{bmatrix} v_{i+1,1}^{j-1} \\ v_{i+1,2}^{j-1} \end{bmatrix} \\ - (M_{2i}^{+})^k\begin{bmatrix} v_{i,1}^{j-1} \\ v_{i,2}^{j-1} \end{bmatrix}\Bigg) \end{aligned} \tag{23}$$

In addition, the restriction operator, $I_{j-1}^{j} = 2(I_j^{j-1})^T$ is used.

Remark: It is clear that the number of required arithmetic

operations to construct this operator is more expensive than the standard operator. However, this is not so important because, the inter-grid transfer operators are only constructed once. The expense of running some systems by the multi-grid Beam-Warming scheme involving this new inter-grid transfer operator is much lower than for the ones involving standard operators.

Now, we want to give an extension of the above inter-grid transfer operator to a 2- dimensional case. To this end, let us consider (●) as the coarse grid points (see Fig. 1). We note that any vector $\mathbf{u} = \begin{bmatrix} A \\ uA \end{bmatrix}$ is defined on the grid points. A prolongation operator from coarse level $j-1$ to the fine level j can be given by:

$$I_{j-1}^{j} v^{j-1}(x,y) =$$

$$\begin{cases} v^{j-1}(x,y) & for \ \bullet \\ \frac{1}{2}\left[v^{j-1}(x,y+\Delta y) + v^{j-1}(x,y-\Delta y)\right] & for \ \blacksquare \\ \frac{1}{2}\left[v^{j-1}(x+\Delta x,y) + v^{j-1}(x-\Delta x,y)\right] & for \ \blacktriangle \end{cases}$$

Also, constructing a non-standard mapping on (○) can be done in a way similar to Eq. (23).

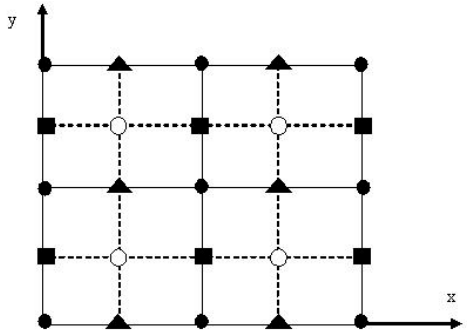

Figure 1

A fine grid with symbols indicating the bilinear interpolation used for the transfer from the coarse grid (●)

Gauss-Seidel block-wise type iteration (relaxation)

In this subsection, we state a smoothing iteration for the Beam-Warming scheme. First, we note that the structure of matrix G in the Beam-Warming scheme (Eq. (18)) is as follows:

$$G = \begin{bmatrix} \times & & & & & & & & & \\ & \times & & & & & & & & \\ \times & \times & \times & \times & \times & \times & & & & \\ \times & \times & \times & \times & \times & \times & & & & \\ & & \times & \times & \times & \times & \times & \times & & \\ & & \times & \times & \times & \times & \times & \times & & \\ & & & & \cdots & \cdots & \cdots & & & \\ & & & & \cdots & \cdots & \cdot & & & \\ & & & & \cdots & \cdots & \cdots & & & \\ & & & & & \times & \times & \times & \times & \times \\ & & & & & \times & \times & \times & \times & \times \\ & & & & & & & \times & \times & \times \\ & & & & & & & \times & \times & \times \\ & & & & & & & & & \times \end{bmatrix}$$

Therefore, the matrix G can be shown

$$G = \begin{bmatrix} D_1 & & & & & & \\ E_2 & D_2 & F_2 & & & & \\ & E_3 & D_3 & F_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & E_{n-1} & D_{n-1} & F_{n-1} & \\ & & & & & D_n & \end{bmatrix}$$

where: $D_i, i = 1,2,\ldots,n$, and E_i, $F_i, i = 2,3,\ldots,n$ are 2 x 2 matrices.

Let us to denote $X = [X_1, X_2, \cdots, X_n]^T$ and $b = [b_1, b_2, \cdots, b_n]^T$ which X_i nd b_i are 2 x 1 vectors. Now, in order to apply the smoothing iterations for Beam-Warming scheme, one can use the Gauss-Seidel block-wise iteration for the system $GX = b$ by:

$$X^{(l)} = X^{(l-1)} + (D-E)^{-1}\left(b - GX^{(l-1)}\right), \quad l = 1,2,\cdots \quad (24)$$

where:

$$D = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{bmatrix}, \quad E = \begin{bmatrix} 0 & & & & \\ -E_2 & 0 & & & \\ & -E_3 & 0 & & \\ & & \ddots & \ddots & \\ & & & -E_{n-1} & 0 \\ & & & & 0 \end{bmatrix}.$$

We note that $D - E$ is a lower triangular bi-diagonal matrix and its inverse is easily computable. A simplified form of the above Gauss-Seidel block-wise iteration for the Beam-Warming scheme reads as follows:

Gauss-Seidel block-wise algorithm for Beam-Warming scheme

1. $X^{(0)} = 0$
2. for $l = 1,2,\cdots$
3. $X_1^{(l)} = D_1^{-1} b_1$
4. $X_n^{(l)} = D_n^{-1} b_n$
5. for $i = 2,3,\cdots,n-1$
6. $X_i^{(l)} = D_i^{-1}\left(F_i X_{i+1}^{(l-1)} - E_i X_{i-1}^{(l)}\right)$
7. end
8. end

Now, in order to clarify multi-grid algorithm for Beam-Warming scheme, we present the 2-grid algorithm as follows:

Two-grid algorithm for Beam-Warming scheme (system of $G_1 X = b$)

1. Set $X^{(0)} = 0$
2. Define $X^{(l)}$ for $l = 1,\cdots,m$ by

$$X^{(l)} = X^{(l-1)} + (D-E)^{-1}\left(b - G_1 X^{(l-1)}\right).$$

3. Set $r_1 = b - G_1 X^{(m)}$.
4. $r_0 = I_1^0 r_1$ (transformation of the residual from the fine level to the coarse level)
5. Solve $G_0 e_0 = r_0$ directly (i.e. $e_0 = G_0^{-1} r_0$) (solving the system at the coarse level directly)
6. Set $e_1 = I_0^1 e_0$. (transformation of the solution in the coarse grid to the fine grid)
7. $X^{(m+1)} = X^{(m)} + e_1$. (modifying of the initial solution in the fine grid)
8. Define $X^{(l)}$ for $l = m+1,\cdots,2m$ by:

$$X^{(l+1)} = X^{(l)} + (D-E)^{-1}\left(b - G_1 X^{(l)}\right)$$

In the above algorithm, as already mentioned, G_0 and G_1 denote the system produced by the Beam-Warming scheme in the coarse and fine grids, respectively. We recall that other relaxation schemes such as incomplete LU factorisation can be applied here.

Available on website http://www.wrc.org.za
ISSN 0378-4738 (Print) = Water SA Vol. 37 No. 2 April 2011
ISSN 1816-7950 (On-line) = Water SA Vol. 37 No. 2 April 2011

Numerical applications

In order to evaluate the accuracy of the block multi-grid algorithm for the Beam-Warming scheme, we first solved a classic open-channel shock wave propagation problem. The numerical properties were analysed and illustrated by 2 different examples. Both examples were solved by Matlab 7.5. Since in the high levels (level 3 upwards), matrix G (which it is obtained by the Beam and Warming scheme) cannot be stored in the computer, we stored matrix G by some block matrices whose numbers depend on the level they are in. Therefore, there is no explicit matrix G and there are merely a number of block matrices, the combinations of which would create matrix G.

In both examples, the V-cycle algorithm is used, in which the number of iterations in pre- and post-smoothing processes has been taken as $m = 2$ for each level j.

Example 1

The hydraulic events following the sudden closure of a gate in a channel are important for designing power channels. For example, the gates of a power channel may be closed instantaneously, which causes an increase in the flow depth. In these situations, the knowledge of the height of the resulting surge is essential for designing a channel.

In this example, the performance of the multi-grid Beam and Warming (MGBW) scheme for surge propagation due to sudden closure of a gate is examined in comparison with the exact solution. We considered a channel with a rectangular cross-section, the bottom of which is 6.1 m wide. The initial conditions in the channel are: 5.79 m deep with a steady discharge of 126 $m^3 \cdot s^{-1}$. The channel is horizontal and frictionless with a length of 5 km. The water surface level in the reservoir is constant at the upstream end and the sluice gate at the downstream end of the channel is suddenly closed at time $t = 0$.

Figure 2 shows the simulated results by the MGBW scheme with the exact solution (Featherstone and Nalluri, 1995) just 90 s after gate closure for different levels. The finest level is considered J=8, and also at the coarsest level, the grid size in space Δx and in time Δt are taken as 25 m and 2.2514 s, respectively. Hence, the system produced by the Beam-Warming scheme at the coarsest level (J=1) has 202 x 202, and at the finest level (J=8) 25 602 x 25 602 dimensions at any time level. It can be seen that the MGBW scheme is in a good agreement with the exact solution. Figure 3 shows the simulated results by the finite-difference Beam and Warming (FDBW) scheme (Chaudhry, 1993) with the exact solution. In this figure, we considered (a) $\Delta x = 25$ and $\Delta t = 2.2514$, (b) $\Delta x = 12.5$ and $\Delta t = 1.1257$, (c) $\Delta x = 6.25$ and $\Delta t = 0.5629$, (d) $\Delta x = 3.125$ and $\Delta t = 0.2814$. Figures 3a, 3b, 3c and 3d correspond to levels 1, 2, 3 and 4 of MGBW scheme, respectively. In this example, the simulated results are satisfactory and hence this scheme is valid. Let and be the approximated solution obtained by MGBW and exact solution, respectively. Table 1 shows the Euclidean error norm $\|\tilde{x} - x\|_2$ for different times from level j=3 to j=8. As we can see, when the level increases, the value of error decreases significantly. In addition, we examined this example without the multi-grid scheme. The CPU time concerning the performance in Example 1 by the MGBW and FDBW schemes at 200 s is given in Table 2. The stop criterion for FDBW is $\|\tilde{x} - x\|_2 \leq 10^{-5}$. In Table 2, MGBW and SMG denote the Beam-Warming and standard inter-grid transfer operators applied to the Gauss-Seidel algorithm, respectively. The standard inter-grid transfer operator that we used is the injection operator. As we observe, the CPU time decreases significantly from level 4 to the end when we use multi-grid scheme. Moreover, in the second and third columns we see that the MGBW algorithm is run faster than the SMG algorithm. Table 2 shows the efficiency of the multi-grid algorithm, clearly.

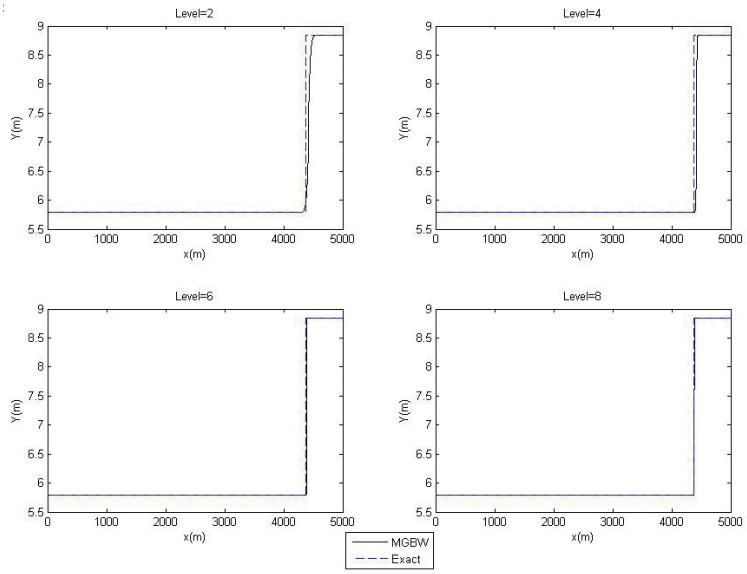

Figure 2
Comparison of simulated results (MGBW scheme) and exact solution for upstream propagating surge

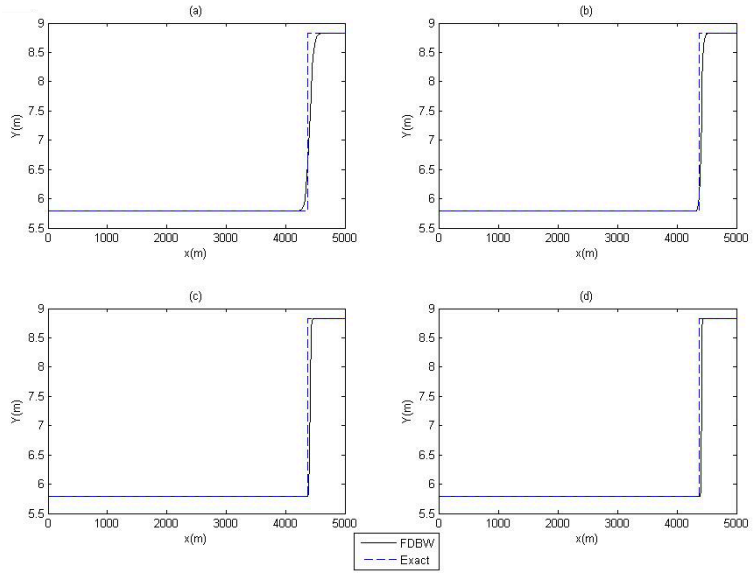

Figure 3
Comparison of simulated results (FDBW scheme) and exact solution for upstream propagating surge

Table 1 Results concerning the performance of the MG algorithm with respect to mesh size			
Level j	t = 100 s	t = 150 s	t = 200 s
$j = 3$	2.4361e-02	6.5627e-02	1.2447e-03
$j = 4$	3.4622e-04	8.5918e-04	1.2550e-03
$j = 5$	3.3473e-06	9.5326e-06	2.4411e-05
$j = 6$	1.0322e-09	7.4017e-09	5.2840e-08
$j = 7$	4.4675e-12	9.8210e-11	1.9052e-10
$j = 8$	6.5609e-14	1.2322e-13	2.0201e-13

Table 2 The CPU time concerning Example 1 with and without multi-grid algorithm at 200 s			
Level j	FDBW	MGBW	SMG
$j = 4$	2.3510e+003	700.8330	745.3467
$j = 5$	4.6852e+003	2.3312e+003	2.6713e+003
$j = 6$	7.30437e+003	3.6047e+003	3.9542e+003
$j = 7$	1.4402 e+004	9.2874e+003	9.6466e+003
$j = 8$	2.2064e+004	1.6467e+004	1.9452e+004

Example 2

Because flood waves normally have a rising limb, a falling limb and a single peak similar to a triangular hydrograph, the latter is usually used as an input function for flood routing analysis. This example is presented for a case of subcritical flow in a rectangular channel with the length of L = 20 km, width of 6 m, uniform bottom slope of S_0 = 0.00193 and a roughness defined by the Manning n_c = 0.025 $m^{-1/3}$ s. Also, uniform flow with a depth of 0.5 m and velocity corresponding to Froude number F_r = 0.5 are considered for the initial conditions. The upstream boundary conditions are defined by a triangular hydrograph $Q = Q(t)$ which changes linearly from 3 to 21.57 $m^3 \cdot s^{-1}$ in 600 s and then decreases to 3 $m^3 \cdot s^{-1}$, again in 600 s. The downstream boundary conditions are: $h = h(t) = const$.

In Fig. 4, $Q = Q(t)$ for the sections x = 1 000 and 2 000 m at the level j=3 with Δx = 10 m and Δt = 3 s, is given by the MGBW and FDBW schemes. Hence, the system resulting from the Beam-Warming scheme at the coarsest level (J=1) has 1002 x 1002, and at the finest level (J=8), 128002 x 128002 dimensions at any time level. The CPU time for Example 2 by the MGBW and FDBW schemes at 200 s is given in Table 3. We considered n=500 iterations for the FDBW scheme in any time. Table 3 shows that the MGBW scheme is run faster than the FDBW scheme.

Table 3 The CPU time for Example 2 by the multi-grid algorithm and finite difference scheme at 200 s		
	FDBW	MGBW
Level	CPU	CPU
$j = 4$	9.2510e+003	2.1746e+003
$j = 5$	1.6852e+004	6.9285e+003
$j = 6$	5.3043e+004	1.3908e+004
$j = 7$	8.8402 e+004	5.0195e+004
$j = 8$	2.2964e+005	8.4840e+004

Conclusions

This paper presents a numerical analysis of the MGBW scheme applied to the complete Saint-Venant equations written for 1-dimensional unsteady flow. Solving the Saint-Venant problem by the Beam and Warming scheme concludes a linear system $Gx = b$. In order to find a good solution, we need to use a very small size for the grid in terms of space (Δx) and time (Δt). Then, the dimension of G would be very large and saving this matrix in a computer is impossible. Hence, most of the iterative methods cannot be used. MGBW has been used to solve the problem. The main factor that influences the convergence rate of the multi-grid continuation algorithms is the distance between the solutions provided by different meshes. MGBW significantly reduces the computational time. The MGBW scheme may be expected to give accurate results for other unsteady open-channel flow problems, such as those involving propagation of a flood wave in a natural stream or flow in a transition.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments and suggestions. Also, the second author gratefully acknowledges the financial support by the Vali-e-Asr University of Rafsanjan, Iran (Grant No. p/1453).

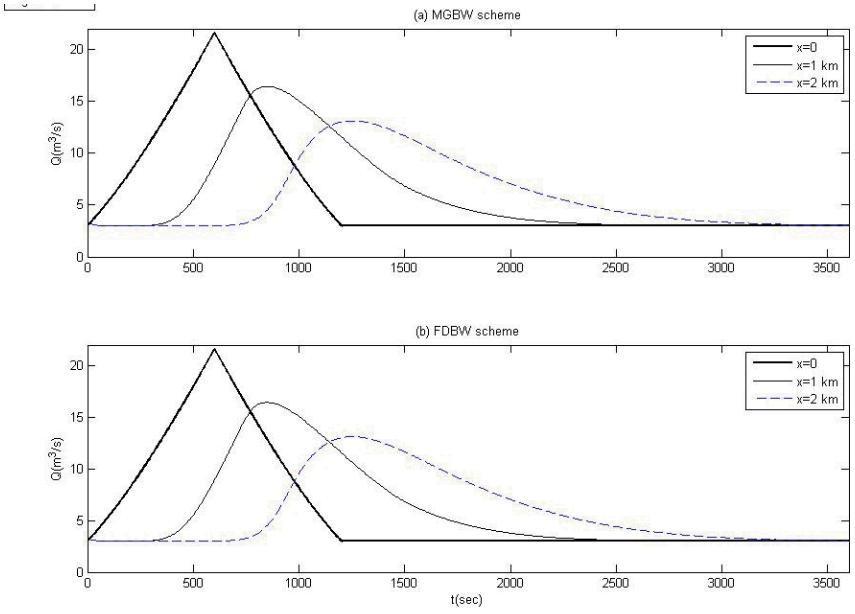

Figure 4
Simulated results for discharge hydrograph

Available on website http://www.wrc.org.za
ISSN 0378-4738 (Print) = Water SA Vol. 37 No. 2 April 2011
ISSN 1816-7950 (On-line) = Water SA Vol. 37 No. 2 April 2011

References

ABBOTT MB (1979) *Computational Hydraulics.* Pitman Publishing, London.

AURELI F, MARANZONI A, MIGNOSA P and ZIVERI CA (2008) Weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Adv. Water Resour.* **31** 962-974.

BEAM RM and WARMING RF (1976) An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *J. Comput. Phys.* **22** 87-110.

BRAMBLE J and PASCIAK J (1987) New convergence estimates for multigrid algorithms. *Math. Comput*. **49** 311-329.

BRAMBLE J, PASCIAK J and XU J (1991) The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms. *Math. Comp.* **56** 1-34.

BRAMBLE JH (1993) *Multigrid Methods.* Pitman, London.

CATELLA1 M, PARIS E and SOLARI L (2008) Conservative scheme for numerical modeling of flow in natural geometry. *J. Hydraul. Eng.* **134** (6) 736-748.

CHAUDHRY MH (1993) *Open-Channel Flow.* Prentice-Hall, Englewood Cliffs, NJ.

CHOI GW and MOLINAS A (1993) Simultaneous solution algorithm for channel networks modeling. *Water Resour. Res*. **29** (2) 321-328.

CHOW V (1959) *Open Channel Hydraulics.* McGraw-Hill, Auckland.

FEATHERSTONE RE and NALLURI C (1995) *Civil Engineering Hydraulics.* Blackwell Science Ltd, Oxford.

JHA AK, AKIYAMA J and URA M (1994) Modeling unsteady open-channel flows modification to Beam and Warming scheme. *J. Hydraul. Eng.* **120** (4) 461-476.

NGUYEN QK and KAWANO H (1995) Simultaneous solution for flood routing in channel networks. *J. Hydraul. Eng.* **121** (10) 744-750.

SEN DJ and GARG NK (2002) Efficient algorithm for gradually varied flows in channel networks. *J. Irrig. Drain. Eng.* **128** (6) 351- 357.

TAVAKOLI A and KERAYECHIAN A (2007) Improvement of the rate of convergence estimates for multigrid algorithm. *App. Math. Comput.* **187** 922-928.

TAVAKOLI A (2010) Convergence of multigrid algorithms in divergence free space for Stokes problem. *Int. J. Appl. Math. Mech.* **6** (3) 62-76.

TROTTENBERG U, OOSTERLEE CW and SCHULLER A (2001) *Multigrid*. Academic Press, New York.

TSENG MH and CHU CR (2000) The simulation of dam-break flows by an improved predictor-corrector TVD scheme. *Adv. Water Res.* **23** 637-643.

VENUTELLI M (2002) Stability and accuracy of weighted four-point implicit finite difference schemes for open channel flow. *J. Hydraul. Eng.* **128** (3) 281-288.