[3] Herrera, I., "Unified Approach to Numerical Methods", Part II, Num. Methods for Partial Differential Equations 1(3), 159-186, 1985.

[4] Herrera, I., Chargoy, L. and Alduncin, G., "Unified Approach to Numerical Methods", Part III, Num. Methods for Partial Differential Equations 1(4), 241-258, 1985.

[5] Herrera, I., "Unifying Concepts in Applied Mathematics", In "The Merging of Disciplines: New Directions in Pure, Applied and Computational Mathematics", Eds. R.E. Ewing, K.I. Gross and C.F. Martin. Springer-Verlag, New York, 1986 (Invited Lecture).

[6] Celia, M.A., T.F. Russell, I. Herrera and R.E. Ewing, "An Eulerian-Lagrangian Localized Adjoint Method for the Advection-Diffusion Equation", Advances in Water Resources, 1990 (In press).

[7] Herrera, I., R.E. Ewing, T.F. Russell and M.A. Celia, Eulerian-Lagrangian Localized Adjoint Methods: Theoretical Foundations", In preparation, 1990.

[8] Allen, M.B., I. Herrera and G.F. Pinder, "Numerical Modeling in Science and Engineering", John Wiley, New York, 1988.

[9] Baptista, A.M., "Solution of advection dominated transport by Eulerian-Lagrangian methods using the backwards methods of characteristics", Ph.D. Thesis, Dept. of Civil Engineering, M.I.T., 1987.

[10] Neuman, S.P., "An Eulerian-Lagrangian numerical scheme for the dispersion convection equation using conjugate space-time grids", J. Comp. Phys., 41, 270-294, 1981.

[11] Neuman, S.P., "Adaptive Eulerian-Lagrangian finite element method for advection-dispersion", Int. J. Num. Meth. Engrg., 20, 321-337, 1984.

[12] Ewing, R.E., T.F. Russell and M.F. Wheeler, "Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics", Comp. Meth. Appl. Mech. Engrg., 47, 73-92, 1984.

[13] Douglas, J. Jr. and T.F. Russell, "Numerical methods for convection dominated diffusion problems based on combining the method of characteristics with finite elemnt or finite difference procedures", SIAM J. Num. Anal., 19, 871-885, 1982.

[14] Espedal, M.S. and R.E. Ewing, "Characteristic Petrov-Galerkin subdomain methods for two phase immiscible flow", Comp. Meth. Appl. Mech. Engrg., 64, 113-135, 1987.

[15] Dahle, H.K., M.S. Espedal, R.E. Ewing and O. Saevareid, "Characteristic adaptive sub-domain methods for reservoir flow problems", Numerical Methods for Partial Differential Equations, to appear, 1990.

A Time-Accurate Multigrid Algorithm for the Solution of the Transport Equations

M. Putti(*), W.A. Mulder(**), W.W-G. Yeh(***)

(*) Dip. Metodi e Modelli Matematici per le Scienze Applicate, University of Padova, Italy

(**) Koninkijke/Shell Exploratie en Produktie Laboratorium, Rijswijk, The Netherlands

(***) Dept. of Civil Engineering, University of California, Los Angeles, USA

ABSTRACT

Explicit time stepping techniques are seldom employed for the numerical solution of transport equations. Severe limitations on the size of the time step are in fact imposed by stability requirements. Using concept drawn from the theory of the Multigrid method, together with localized mesh refinements, an algorithm is described that allows the use of large time steps in explicit schemes without violating any stability criterion.

Starting from a fine grid, the solution after few time steps is interpolated to coarser grids where larger time steps can be used. An estimate of the local time-discretization error is then obtained by comparison of the solution at different grid levels and different times. To maintain fine grid accuracy, the solution is interpolated back to finer grids only in the parts of the domain where the time-discretization errors are deemed too large. The scheme therefore progresses simultaneously on different levels in different parts of the domain. Time stepping on fine grids is performed only over small subsets of the domain where accuracy is low. On most part of the domain very coarse grids, and consequently large time steps, can be used. In this manner, time steps that are on the average comparable to those used in implicit schemes can be employed, and fine grid accuracy is maintained throughout the entire domain.

INTRODUCTION

In the numerical solution of advection-dispersion equations by means of finite difference or finite element methods the time derivatives are usually discretized using explicit or implicit schemes. Explicit techniques do not require the solution of systems of algebraic equations, so that each time evolution step is computa-

tionally cheap. However, numerical stability imposes severe limitations on the size of the admissible time step. The procedure becomes therefore extremely expensive when solutions relatively far from initial data are required. On the other hand, implicit schemes do not have to satisfy any dynamic stability requirement: the size of the time step is dictated only by accuracy considerations. A system of algebraic equations has to be solved in this case in order to advance in time. Each step is computationally expensive especially when nonlinear equations are to be solved.

Stability constraints usually depend on the size of the grid spacing: for small grid spacing small time steps must be used. For example, with first order explicit time stepping (forward Euler) and second order central differencing for viscous terms, the admissible time step, according to the von Neumann analysis, is proportional to the square of the minimum grid spacing employed [1]. This fact suggests that if the computations are performed on sufficiently coarse grids, large time steps may be used.

The goal of these notes is to describe an algorithm based on the multigrid approach which, overcoming stability constraints, allows for the use of large time steps in conjunction with explicit schemes. The multigrid method is a general procedure for the solution of difference equations coming from the discretization of differential problems. It is widely used in conjunction with finite differences or finite elements as a solver for elliptic problems [2]. With an appropriate use of information gathered on computational grids with different spacings, the method achieves great efficiency. The multigrid, as a solver for time dependent problems, is usually employed for the solution of the discrete equations arising at each time step [3]. The idea of using coarser grids in order to achieve larger time steps without incurring in instabilities has been exploited in [4], where a multigrid algorithm which uses typical V-cycles is derived. However the results obtained are not very promising: the solution deteriorates rapidly as time progresses and as the number of grids increases. A few other examples of applications to time dependent equations can be found in [2].

THE TIME-ACCURATE MULTIGRID

The algorithm, whose ideas were first proposed by Brandt ([5]), starts from the following fundamental observation. The use of explicit time differences is limited by the size of the time step dictated by stability conditions especially when parabolic-type equations, like the ones governing the transport of contaminants in groundwater, have to be integrated in time. In this case, high frequency components of the solution change rapidly and reach steady state much sooner than low frequencies. This means that after an initial transient phase, the solution changes smoothly until steady state is reached and low frequencies can be considered the dominant features of the problem.

Numerically this implies that after a few time steps performed on a fine grid, accurate computations can be obtained on increasingly coarser grids by neglecting the changes in the high frequency components of the numerical solution. As a

consequence, large time steps can be employed without violating the stability conditions required by explicit time differencing on finer grids. Whenever the high frequencies cannot be neglected, they must be updated by performing time steps on the finer level.

The problem of deciding when to compute on finer grids is solved by estimating the time-discretization error (TDE) by comparing the results obtained at different times on different levels. When the TDE is larger than a prefixed tolerance, high frequency changes cannot be neglected and computations on finer levels are necessary. Since fine-grid accuracy in space is maintained throughout the calculations by using a coarse-grid correction-type scheme ([6]), the effects of the spatial discretization errors (SDE) can be subtracted from the total discretization error. Accuracy checks are performed for each grid at each time step. To achieve better efficiency this last phase can be done locally, and zones of the computational grid where refinement is needed can be isolated. Adaptive mesh refinement is therefore easily implemented.

THE ALGORITHM

Given a partial differential equation:

$$\frac{\partial u}{\partial t} = r(u) \tag{1}$$

where $r(u)$ will be called the *residual*, a numerical scheme can be written as:

$$\begin{aligned} \frac{u(x_i, t_{n+1}) - u(x_i, t_n)}{t_{n+1} - t_n} - r^\ell(u) &= T\left(u(x_i, t_n), x_i, t_n\right) \\ &= \tau(u, x_i, , t_n) + \sigma(u, x_i, t_n) \end{aligned} \tag{2}$$

where r^ℓ is the discrete residual operator, T is the total discretization error, τ is the spatial discretization error, σ is the temporal part of T, h is the grid spacing, and k is the time step. If $\tau = O(h^p)$ and $\sigma = O(k^q)$ the scheme is said to be of order (p, q) [7].

Some notational conventions must be now defined. The grids (or levels) are identified by numbers. To simplify the notation it is assumed that the coarser grids are all subsets of the finest. This, however, does not have to be absolutely observed and the different grids can in principle be completely different. The finest level is indicated by ℓ and the coarsest by 1. The residual on grid ℓ is denoted by r^ℓ, the numerical solution by u^ℓ. The local estimate of the space truncation error is τ^ℓ, and $\tau_\ell^{\ell-1}$ represents the relative truncation error obtained from calculations performed on grids $\ell - 1$ and ℓ.

Intergrid transfer functions must also be defined as follows. The *restriction* operator moves the information from level ℓ to level $\ell - 1$, and it is indicated by $I_\ell^{\ell-1}$ if it acts on the numerical solution u, and by $\tilde{I}_\ell^{\ell-1}$ if it acts on the numerical residual r^ℓ. The *prolongation* operator, denoted by $I_{\ell-1}^\ell$, interpolates the values

from grid $\ell - 1$ to grid ℓ, and acts always on the solution: $u^{\ell} = I_{\ell-1}^{\ell} u^{\ell-1}$. Both these operator are defined in terms of the space discretization scheme, so that they are consistent with the SDE.

An estimate of $\tau(x,t)$ can be obtained using standard multigrid techniques, namely by the use of the coarse-grid correction scheme ([6]). The algorithm for the determination of the estimate of $\sigma(x,t)$ starts with the calculation of the residual on the finest grid at the initial time t_n: $r^{\ell}(t_n)$. Then the solution u^{ℓ} on grid ℓ can be interpolated to the coarser grid $\ell - 1$, and the residual can be calculated as:

$$\begin{aligned} u^{\ell-1}(t_n) &= I_{\ell}^{\ell-1} u^{\ell}(t_n) \\ r^{\ell-1}(t_n) &= r^{\ell-1}\left[u^{\ell-1}(t_n)\right] \end{aligned} \tag{3}$$

The local spatial truncation error due to the transfer of information from grid ℓ to grid $\ell - 1$ is:

$$\tau_{\ell}^{\ell-1}(t_n) = r^{\ell-1}(t_n) - I_{\ell}^{\ell-1} r^{\ell}(t_n) \tag{4}$$

A modified residual can be now introduced at level $\ell - 1$:

$$\overline{r}^{\ell-1}(t_n) = r^{\ell-1}(t_n) + \tau_{\ell}^{\ell-1}(t_n) \tag{5}$$

Note that in $\overline{r}^{\ell-1}$ only the effects of the spatial discretization are copensated. With similar steps the modified residual can be evaluated also at level $\ell - 2$:

$$\overline{r}^{\ell-2}(t_n) = r^{\ell-2}(t_n) + \tau_{\ell}^{\ell-2}(t_n) \tag{6}$$

where now $\tau^{\ell-1}(t_n) = \tau_{\ell-1}^{\ell-2} + \tilde{I}_{\ell-1}^{\ell-2} \tau_{\ell}^{\ell-1}$. The procedure neglects the temporal part of the truncation error: this approximation is acceptable as long as σ remains small.

The estimation of σ proceeds as follows. Let S be the discrete evolution operator. Let $\overline{u}$ denote the numerical solution integrated with the modified residual $\overline{r}$:

$$\overline{u}^{\ell}(t_{n+1}) = \overline{u}^{\ell}(t_n) + (t_{n+1} - t_n)\overline{r}^{\ell}(t_n) \tag{7}$$

Starting with (3),(4), (5), and (6), we obtain:

$$\begin{aligned} \overline{u}^{\ell-2}(t_{n+1}) &= S^{\ell-2}\left(u^{\ell-2}(t_n), \overline{r}^{\ell-2}(t_n), \tau_{\ell}^{\ell-2}(t_n)\right) \\ \overline{u}^{\ell-1}(t_{n+1}) &= S^{\ell-1}\left(u^{\ell-1}(t_n), \overline{r}^{\ell-1}(t_n), \tau_{\ell}^{\ell-1}(t_n)\right) \end{aligned} \tag{8}$$

The relative temporal truncation error is estimated as:

$$\sigma_{\ell-1}^{\ell-2}(t_{n+1}) = \frac{\overline{u}^{\ell-2}(t_{n+1}) - I_{\ell-1}^{\ell-1}\overline{u}^{\ell-1}(t_n + 1)}{t_{n+1} - t_n} \tag{9}$$

The calculation can continue on coarser levels if $\sigma_{\ell-1}^{\ell-2}$ remains smaller than a fixed tolerance ϵ. If after k time steps $\sigma_{\ell-1}^{\ell-2} > \epsilon$, then the temporal part of the truncation error, that was neglected in (5) and in (6), must be updated by performing the calculations at the finest level ℓ.

It is easy to see now how the same algorithm can be applied with more levels. If $\sigma_{\ell-1}^{\ell-2}$ is smaller than a tolerance, (desired accuracy), the scheme can progress on the coarser grid $\ell - 3$ and accuracy checks can be performed on $\sigma_{\ell-2}^{\ell-3}$ instead of $\sigma_{\ell-1}^{\ell-2}$, and so on. If the solution change remain smooth enough, grids that are many times coarser than the original can be used, so that considerable time is saved in the calculations. However if the solution is not smooth in certain part of the domain the algorithm forces the calculations always at the finest level. Since generally the non smooth part of the solution is localized, the problem is solved identifying the parts of the domain where σ is too large, and performing local refinements on those areas. In this manner time stepping takes place on the finer grids only over small subsets of the numerical domain, while the computations are performed mostly on coarser levels in such a way that, on the average, large time steps can be used. Hence the scheme uses more than one grid at the same time and adaptively chooses the appropriate levels performing the calculations on the smallest set of nodes compatible with the desired accuracy.

IMPLEMENTATION OF THE ALGORITHM

The algorithm has been implemented for the solution of the advection-dispersion equation in one dimension. The standard three point stencil is used for the dispersive fluxes while central differences are used for the advective terms. The forward Euler scheme is implemented to advance in time; the stability condition is therefore the classical von Neumann criterion $2Dk/h^2 < 1$, where D is the dispersion coefficient. Coarsening or refinement of the grids is performed locally, only in those part of the computational domain where $\sigma_{\ell-1}^{\ell-2}$ of equation (9) exceeds a fixed tolerance ϵ. At any level the local grids are prevented from being too small, i.e. the parts of the grids that are formed by less than a predetermined number of nodes are collapsed into the next finer grid. If $\sigma_{\ell-1}^{\ell-2}$ is smaller or larger than prefixed tolerances, respectively coarsening or refinement is performed.

A few numerical examples are run to demonstrate the applicability of the proposed algorithm. The problem that is simulated is a purely diffusive case with constant diffusion coefficient and a unit pulse at $x = 0.5$ as initial conditions. The boundary conditions are of the Dirichlet type, and their numerical values correspond to the analytical solution to the same problem at corresponding times. The finest grid is discretized with $2^7 + 1 = 129$ nodes. The maximum number of levels that is used is 4 and 5 so the coarsest grids have respectively $2^4 + 1 = 17$ and $2^3 + 1 = 9$ nodes. The results are compared with those obtained with a one grid code using the same finite difference scheme. In table 1 are shown the values of the average time step size used by the two codes and of the L_∞ and L_2 norms of the errors (difference between numerical and analytical solutions). In figure 1

n. of levels TDMGM code	n. of nodes 1 grid code	average k	L_∞ norm of errors	L_2 norm of errors
–	129	0.18×10^{-3}	0.14×10^{-4}	0.93×10^{-5}
–	17	0.12×10^{-1}	0.54×10^{-3}	0.25×10^{-1}
–	9	0.47×10^{-1}	0.13×10^{-2}	0.10
4	–	0.59×10^{-2}	0.43×10^{-4}	0.18×10^{-4}
5	–	0.16×10^{-1}	0.61×10^{-4}	0.35×10^{-4}

Table 1: Comparison between the TDMGM code and the one-grid code

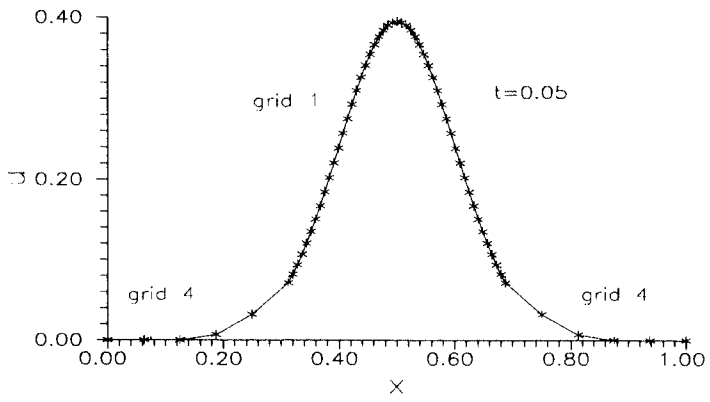


Figure 1: Numerical solution as calculated from the TAMGM code. The effect of the automatic refinement algorithm is visible.

the results of the calculations are plotted at $t = 0.05$. It can be noted that the TAMGM scheme uses the finest level (129 nodes) to calculate the solution where more accuracy is required. In the two parts of the domain close to the boundaries where the solution is smoother the algorihm uses the coarser grids. Note also that the time step required by the multigrid code is many times bigger than that required by the one grid code. The accuracy deteriorates slowly as the number of levels increases.

CONCLUSIONS

The use of large time steps in conjunction with explicit schemes has been obtained by an appropriate utilization of concepts drawn from the multigrid method. Numerical results show that the algorithm, while maintaining the desired accuracy, is able to use time step sizes that are many times bigger than those admissible for the one-grid code. Because of the complexity of the adaptive refinement technique, the TAMGM does not improve the efficiency of the one-grid scheme in terms of CPU time. However, in more than one spatial dimension, the scheme should improve its efficiency since the number of nodes decreases more substantially in passing form fine to coarse levels. Furthermore, the use of explicit schemes avoids the solution of systems of algebraic equations. This feature can dramatically improve the efficiency of the scheme in the solution of nonlinear problems.

References

[1] Peyret, R. and T. D. Taylor. *Computational Methods for Fluid Flow.* Springer-Verlag, New York, 1983.

[2] Hackbusch, W. *Multi–Grid Methods and Applications.* Springer–Verlag, New York, 1985.

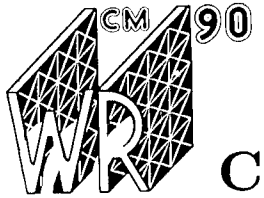
[3] Schröder, W. and D. Hänel. A Comparison of Several MG-Methods for the Solution of the Time-Dependent Navier Stokes Equations. In Hackbusch, W. and U. Trottenberg, editors, *Multigrid Methods II*, Springer–Verlag, New York, 1986.

[4] Jespersen, D. C. A Time–Accurate Multiple–Grid Algorithm. *AIAA* 7th *Computational Fluid Dynamics Conference, paper no. 85–1493–CP*, 1985.

[5] Brandt, A. Multi-Level Adaptive Finite-Element Methods. I. Variational Problems. In J. Freshe, D. Pallschke, U. T., editor, *Special Topics of Applied Mathematics*, North-Holland, 1980.

[6] Briggs, W. L. *A Multigrid Tutorial.* SIAM Publications, Philadelphia, 1987.

[7] Richtmeyer, R. D. and K. W. Morton. *Difference Methods for Initial–Value Problems.* Interscience, New York, 1967.



Computational Methods in Surface Hydrology

Proceedings of the Eighth International Conference on Computational Methods in Water Resources, held in Venice, Italy, June 11-15 1990.

(Vol. 2)

Editors: G. Gambolati
A. Rinaldo
C.A. Brebbia
W.G. Gray
G.F. Pinder

Computational Mechanics Publications, Southampton Boston

Co-published with

Springer-Verlag, Berlin Heidelberg New York London Paris Tokyo