INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2002; **53**:1301–1322 (DOI: 10.1002/nme.329)

Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow

Dmitri Kavetski, Philip Binning*,† and Scott W. Sloan

Department of Civil, Surveying and Environmental Engineering, University of Newcastle, Callaghan, NSW 2308, Australia

SUMMARY

An automatic time stepping scheme with embedded error control is developed and applied to the moisture-based Richards equation. The algorithm is based on the first-order backward Euler scheme, and uses a numerical estimate of the local truncation error and an efficient time step selector to control the temporal accuracy of the integration. Local extrapolation, equivalent to the use of an unconditionally stable Thomas–Gladwell algorithm, achieves second-order temporal accuracy at minimal additional costs. The time stepping algorithm also provides accurate initial estimates for the iterative non-linear solver. Numerical tests confirm the ability of the scheme to automatically optimize the time step size to match a user prescribed temporal error tolerance.

An important merit of the proposed method is its conceptual and computational simplicity. It can be directly incorporated into existing or new software based on the backward Euler scheme (currently prevalent in subsurface hydrologic modelling), and markedly improves their performance compared with simple fixed or heuristic time step selection. The generality of the approach also makes possible its use for solving PDEs in other engineering applications, where strong non-linearity, stability or implementation considerations favour a simple and robust low-order method, or where there is a legacy of backward Euler codes in current use. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: Richards equation; adaptive time stepping; backward Euler scheme

1. INTRODUCTION

Accurate and efficient modelling of unsaturated flow is of great importance in environmental science and engineering. Unsaturated flow models are routinely used in hydrology to estimate infiltration and runoff. In hydrogeology, these models serve to analyse the dynamics of the surface–aquifer interface and to estimate contaminant migration. Prediction of soil moisture variations is critical in analysing the behaviour of reactive soils, which is an important infrastructure issue in semi-arid locations such as Australia. High-profile areas of engineering,

*Correspondence to: Philip Binning, Department of Civil, Surveying and Environmental Engineering, University of Newcastle, Callaghan, NSW 2308, Australia
†E-mail: pbinning@mail.newcastle.edu.au

Received 24 July 2000
Revised 4 January 2001

Copyright © 2001 John Wiley & Sons, Ltd.

including nuclear waste repository design, are also dependent on accurate models of partially saturated flows. Other uses include geochemical and agricultural applications.

Richards equation is typically used to describe unsaturated flows (e.g. References [1–3]), and is derived from the continuity and Darcy equations. It is a highly non-linear PDE that can be cast in several forms, depending on whether pressure, moisture, or both are used as state variables. The moisture form of Richards equation is given by

$$\frac{\partial \theta}{\partial t} - \nabla \cdot D(\theta) \nabla \theta + \frac{\partial K(\theta)}{\partial z} = 0 \tag{1}$$

where θ is the volumetric moisture content [−], z is the (positive downward) depth [L], t is time [T], $K(\theta)$ is the unsaturated hydraulic conductivity [L/T] and $D(\theta)$ is the unsaturated diffusivity [L^2/T].

The highly non-linear dependence of the hydraulic conductivity and diffusivity on the moisture content, in combination with the non-trivial forcing conditions that are often encountered in engineering practice, makes Richards equation virtually intractable to analytic approaches. The practical utility of analytic and semi-analytic solutions is limited by their restrictive assumptions, which, most notably, are homogeneity of the soil medium and a simple mathematical form for the constitutive and forcing functions. As a result, a multitude of numerical algorithms, typically based on low-order finite difference or finite element schemes, have been proposed to approximate Richards equation (e.g. References [4, 5]).

The spatial approximation of Richards equation is usually accomplished using finite element or finite difference methods (e.g. References [4, 6]). Mass lumping is normally employed to improve the numerical stability of finite element models [4, 7]. An adaptive finite element algorithm has also been investigated [8].

The solution of the non-linear algebraic systems that arise in implicit numerical discretizations of Richards equation has been the subject of significant research. Iterative schemes (e.g. Picard, Newton, Newton–Krylov, fast-secant and relaxation methods), as well as non-iterative methods (e.g. the implicit factored scheme) have been proposed [6, 9–13]. In practice, the fixed-point (Picard) iteration scheme is prevalent due to its simplicity and satisfactory performance (e.g. References [4, 9]). The treatment of the constitutive functions has also been addressed, demonstrating improved solver performance when smooth interpolation is used [14].

This paper focuses on the appropriate handling of numerical errors that are associated with the temporal approximation of unsaturated moisture flows. Previous studies indicate the significance of this aspect for reliable numerical simulations. For example, Celia *et al.* [4] and Rathfelder and Abriola [15] highlighted the importance of adequate time approximation for the conservation of mass in the pressure form of Richards equation. Empirical assessment of simple time stepping schemes, including the backward Euler and Crank–Nicolson schemes, showed that second-order schemes are generally more efficient than first-order schemes, although first-order schemes are more cost-effective at coarse stepsizes [9, 16]. Other time stepping schemes used for Richards equation include the three-level Lees' scheme [9], the Douglas–Jones predictor–corrector method [17], implicit Runge–Kutta schemes [18] and backward-difference formulae [5].

However, a major weakness of most time stepping algorithms applied to Richards equation is their approach to stepsize selection. Indeed, the two most common time stepping strategies are:

Copyright © 2001 John Wiley & Sons, Ltd.

(a) Uniform time increments (e.g. References [4, 9] and most other studies). This technique tacitly assumes that the behaviour of the solution is constant throughout the simulation. Under variable forcing conditions, the solutions vary widely in character and uniform time stepping becomes inappropriate;
(b) Heuristic approaches, which vary the stepsize according to the convergence of the non-linear solver (e.g. References [15, 19]). However, the absence of a clear quantitative link between the discretization errors and solver performance yields little, if any, theoretical guidance for optimizing and verifying the heuristic algorithms. Additionally, heuristic techniques lack generality with respect to different non-linear solvers, especially for non-iterative implementations.

These standard approaches for stepsize variation within numerical hydrological simulations are crude, since no attempt is made to rigorously monitor the adequacy of the stepsize for the particular solution behaviour. At the same time, adaptive integration has become standard practice in numerical ODE analysis. As elegantly stated by Press *et al.* [20] (pp. 553, 534):

> We consider adaptive stepsize control … essential for serious computing […]. Many small steps should tiptoe through treacherous terrain, while a few great strides should speed through smooth uninteresting countryside. The resulting gains in efficiency are not mere tens of percents or factors of two; they can sometimes be factors of ten, hundred, or more.

Recently, Tocci *et al.* [5], Miller *et al.* [14] and Williams and Miller [21] described the application of an existing differential-algebraic equation solver (DASPK) to integrate in time the pressure form of Richards equation, with the specific aim of providing automatic time step selection and error control. This paper also presents a technique for improving the temporal integration, but specifically designed for use in practical codes for Richards equation and non-linear PDEs in general, which tend to use simple low-order methods such as the backward Euler scheme.

The DASPK algorithm employed by Tocci *et al.* [5] is a variable-stepsize variable-order (up to fifth) differential-algebraic equation (DAE) solver. Whilst high-order schemes quickly become more efficient than low-order approximations as the error tolerance is decreased, hydrologic and engineering practice rarely demands accuracy finer than about 0.1%. In practical subsurface flow simulations, the uncertainty in the soil hydraulic properties and the forcing conditions usually exceeds numerical approximation errors. At low to medium accuracy requirements, high-order does not necessarily translate into high performance. For example, Wood [16] (p. 264) states that, for a particular test problem, second-order formulations (the Crank–Nicolson scheme) outperformed first-order methods (the backward Euler scheme) only when relative errors below 0.005% were required. The work of Paniconi *et al.* [9] also suggests that first-order schemes are competitive with second-order schemes when coarse time steps are used. In these circumstances, a variable order algorithm may be limited to its low-order formulae, in many cases precisely the backward Euler method. Indeed, the results of Tocci *et al.* [5] indicate that DASPK has a similar efficiency to the backward Euler scheme at low to medium accuracy requirements, and is much more efficient at fine error tolerances, as the order of the approximation is progressively increased.

A limitation of many standard ODE solvers is their inability to handle problems with several state vectors, such as the semi-discrete mixed form of Richards equation. Whilst DAE algorithms can, in principle, integrate the semi-discrete mixed Richards equation, such work

Copyright © 2001 John Wiley & Sons, Ltd.

has not yet been done. In this paper, the general methodology for the new method is presented and applied to the moisture form of Richards equation. In a companion paper [22], the scheme is generalized to the mixed form of the equation, leading to an intrinsically mass conservative algorithm.

Stability constraints, related to the stiffness of the ODEs that arise following the finite element or finite difference spatial approximation of parabolic PDEs such as Richards equation, may also limit the order of accuracy available to an ODE integrator [16, 23]. Approximations of higher order than the governing DE itself are potentially unstable, due to the risk of extraneous solutions [20]. Hence, in practice, it may be prudent to employ robust first- and second-order time stepping schemes, especially if they come with the added benefit of easier code maintenance (due to its simplicity).

In engineering applications, it is desirable to use simple approaches for adaptive time step selection. The algorithm presented in this paper is based on the widely used backward Euler scheme and can therefore be easily incorporated into existing unsaturated flow codes. A variant of this scheme was introduced by Sloan and Abbo [24] for the integration of elastoplastic consolidation equations in geomechanics, and was found to provide appreciable gains in accuracy and efficiency over existing constant-stepsize and heuristic time stepping formulations [25].

This study extends the approach of Sloan and Abbo [24] to the numerical time integration of the moisture-based Richards equation. Although incapable of modelling discrete material profiles or variably saturated/unsaturated flows, the moisture form is inherently mass conservative (unlike the pressure form) and, as it contains a single state variable, is well suited for a clear development of the formulation. The use of a single spatial dimension (depth z) further facilitates the presentation and empirical assessment of the time stepping algorithm. Since we propose the new scheme as an improvement to be incorporated into existing and new backward Euler codes, the performance benchmarking will also focus on these simple but widely used schemes.

2. ALGORITHM DEVELOPMENT

For the numerical solution of PDEs such as Richards equation, it is convenient to decouple the issues of temporal and spatial accuracy by applying a finite element approximation to the spatial operator, while employing time marching methods to integrate the resulting initial-value problem. The proposed formulation treats a spatially discretized form of Richards equation as a system of simultaneous ODEs and controls the temporal errors by applying concepts developed in the field of ODE analysis.

2.1. Spatial approximation

The Galerkin finite element method is widely used for the solution of PDEs such as the moisture form of Richards equation (e.g. Reference [2]). Regardless of the spatial dimensionality of the problem, it leads to the following first-order ODE system, of rank equal to the total number of nodes in the spatial mesh

$$\mathbf{M}\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} + \mathbf{K}\boldsymbol{\theta} = \mathbf{F} \tag{2}$$

Copyright © 2001 John Wiley & Sons, Ltd.

where $\mathbf{M}, \mathbf{K}$ and $\mathbf{F}$ are the global finite element matrices. $\mathbf{M}$ is commonly referred to as the mass (or time) matrix, $\mathbf{K}$ is the conductivity matrix and $\mathbf{F}$ is a generalized forcing vector (which also contains the boundary fluxes). When one-dimensional vertical flows are simulated and linear basis functions are used for both the state variable and the constitutive functions, the elemental matrices are given by

$$m_{i,j}^{(e)} = \int_0^{L(e)} N_i N_j \, dz \;\Rightarrow\; \mathbf{M}^{(e)} = \frac{L^{(e)}}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \Rightarrow \mathbf{M}_L^{(e)} = \frac{L^{(e)}}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

$$k_{i,j}^{(e)} = \int_0^{L(e)} D \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \, dz \;\Rightarrow\; \mathbf{K}^{(e)} = \frac{D^{(e)}}{L^{(e)}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{4}$$

$$f_i^{(e)} = \int_0^{L(e)} K \frac{\partial N_i}{\partial z} \, dz + \left[\left(D \frac{\partial \theta}{\partial z} - K \right) N_i \right]_0^{L(e)} \;\Rightarrow\; \mathbf{F}^{(e)} = K^{(e)} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{5}$$

where $\mathbf{N}$ is the set of basis functions and $L^{(e)}$ is the elemental length. The elemental values of D and K are evaluated using arithmetic means, $D^{(e)} = 1/2(D_1 + D_2)$ and $K^{(e)} = 1/2(K_1 + K_2)$, where the subscripts denote the local node index. The boundary flux q vanishes at all interior nodes and at boundary nodes where Dirichlet conditions (specified moisture content) are imposed.

To improve the numerical stability of (2), the mass matrix $\mathbf{M}$ has been diagonalized (lumped), as shown in (3). Although lumping introduces numerical diffusion, it eliminates undesirable oscillations generated by consistent (unlumped) mass matrices [4].

It is emphasized that any other approach that generates a spatially discrete ODE system similar to (2) could be used. As the current development is concerned with the time integration of Richards equation, the simple choice of spatial discretization given by (2)–(5) is sufficient.

2.2. *Temporal integration*

Applied ODE analysis makes extensive use of *a posteriori* local truncation error estimates obtained by comparing two approximations of adjacent order of accuracy [23]. Alternative approaches to error estimation include comparing solutions obtained with different time step sizes [20] or, less commonly, using *a priori* estimates [26]. The simplest and most common numerical schemes for the solution of (2) belong to the weighted Euler difference family:

$$[\mathbf{M} + \phi \, \Delta t \, \mathbf{K}^{n+\phi}] \mathbf{V}^{n+\phi} = -\mathbf{K}^{n+\phi} \boldsymbol{\theta}^n + \mathbf{F}^{n+\phi} \tag{6}$$

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \Delta t \mathbf{V}^{n+\phi} \tag{7}$$

where $\mathbf{V}^{n+\phi}$ is an $O(\Delta t)$ approximation of $(d\boldsymbol{\theta}/dt)^{n+\phi}$ and $\mathbf{K}^{n+\phi} = \mathbf{K}(\boldsymbol{\theta}^n + \phi \, \Delta t \, \mathbf{V}^{n+\phi})$. The Euler scheme is cast in the form of (6)–(7) to exploit similarities with the Thomas–Gladwell family presented below.

The parameter ϕ varies in the range $0 \to 1$. The ϕ-schemes are unconditionally stable when $\phi \geqslant 1/2$ and are $O(\Delta t)$ accurate, with the exception of the $O(\Delta t^2)$-convergent Crank–Nicolson scheme ($\phi = 1/2$). Setting $\phi = 1$ leads to the backward Euler (fully implicit) scheme, which is only first-order accurate, but very stable and hence ideally suited for the integration of stiff

Copyright © 2001 John Wiley & Sons, Ltd.

ODE systems that arise from finite element or finite difference semi-discretization of parabolic PDEs such as Richards equation [16]. Although formally more accurate, the Crank–Nicolson scheme often suffers from troublesome bounded oscillations induced by abrupt changes in forcing and boundary conditions and is therefore less popular in engineering practice than the backward Euler scheme [16].

A family of approximations, proposed by Thomas and Gladwell [27] for the solution of second-order ODE systems, can also be applied, as a special case, to the first-order system (2). These approximations are given by

$$[\varphi_2 \Delta t \mathbf{M} + \varphi_3 \Delta t^2 \mathbf{K}]\ddot{\boldsymbol{\theta}}^n = -\mathbf{M}\dot{\boldsymbol{\theta}}^n - \mathbf{K}(\boldsymbol{\theta}^n + \varphi_1 \Delta t \dot{\boldsymbol{\theta}}^n) + \mathbf{F}^{n+\varphi_1} \tag{8}$$

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \Delta t \dot{\boldsymbol{\theta}}^n + 1/2\Delta t^2 \ddot{\boldsymbol{\theta}}^n \tag{9}$$

$$\dot{\boldsymbol{\theta}}^{n+1} = \dot{\boldsymbol{\theta}}^n + \Delta t \ddot{\boldsymbol{\theta}}^n \tag{10}$$

where the matrix $\mathbf{K}$ is evaluated as $\mathbf{K}(\boldsymbol{\theta}^n + \varphi_1 \Delta t \dot{\boldsymbol{\theta}}^n + \varphi_3 \Delta t^2 \ddot{\boldsymbol{\theta}}^n)$. The schemes are unconditionally stable provided $2\varphi_3 \geqslant \varphi_1 > 1/2$ and $\varphi_2 \geqslant 1/2$ [27]. When applied to first-order ODEs, the Thomas–Gladwell schemes are $O(\Delta t^2)$ convergent provided $\varphi_1 = \varphi_2$ [16].

While the mass matrix $\mathbf{M}$ is constant for the semi-discrete moisture form of Richards equation, for other PDEs (e.g., the pressure form of Richards equation) $\mathbf{M}$ may be non-linear in the state variable. Both the Euler and Thomas–Gladwell families handle such cases in a straightforward fashion, using intermediate approximations $\mathbf{M}(\boldsymbol{\theta}^n + \phi \Delta t \mathbf{V}^{n+\phi})$ and $\mathbf{M}(\boldsymbol{\theta}^n + \varphi_1 \Delta t \dot{\boldsymbol{\theta}}^n + \varphi_3 \Delta t \ddot{\boldsymbol{\theta}}^n)$, respectively.

The Thomas–Gladwell equations can be re-arranged as follows:

$$[\varphi_2 \mathbf{M} + \varphi_3 \Delta t \mathbf{K}]\dot{\boldsymbol{\theta}}^{n+1} = [-(1-\varphi_2)\mathbf{M}\dot{\boldsymbol{\theta}}^n - (\varphi_1 - \varphi_3)\Delta t \mathbf{K}\dot{\boldsymbol{\theta}}^n] - \mathbf{K}\boldsymbol{\theta}^n + \mathbf{F}^{n+\varphi_1} \tag{11}$$

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + 1/2\Delta t(\dot{\boldsymbol{\theta}}^n + \dot{\boldsymbol{\theta}}^{n+1}) \tag{12}$$

It can be observed that the non-linear systems (6) and (11) are quite similar and, in fact, can be made identical by setting $\phi = \varphi_1 = \varphi_3$ and $\varphi_2 = 1$. In doing so, the Euler and Thomas–Gladwell solution estimates are evaluated after the solution of a single non-linear system, which is a very valuable efficiency advantage. Additionally, the vectors containing the numerical estimates of the first derivative coincide, so that $\mathbf{V}^{n+\phi} \equiv \dot{\boldsymbol{\theta}}^{n+1} = \dot{\boldsymbol{\theta}}^n + \Delta t \ddot{\boldsymbol{\theta}}^n$. These properties simplify the algorithm and lead to a compact and efficient implementation with minimal computer storage.

The combination of the single non-linear system inversion with the $O(\Delta t^2)$ accuracy requirement of the Thomas–Gladwell family limits the choice of weighting parameters to

$$\phi = \varphi_1 = \varphi_2 = \varphi_3 = 1 \tag{13}$$

It is also beneficial that the parameter choice (13) satisfies the unconditional stability requirements of both the Euler difference and Thomas–Gladwell approximations. Unconditional stability is a highly desirable property of practical time stepping algorithms, as it permits the stepsize to be regulated purely by accuracy requirements, unhindered by stability restrictions.

Copyright © 2001 John Wiley & Sons, Ltd.

Selecting the parameter set (13), and noting that $\dot{\boldsymbol{\theta}}^{n+1} \equiv \mathbf{V}^{n+1}$, leads to the following set of equations:

$$[\mathbf{M} + \Delta t\, \mathbf{K}^{n+1}]\dot{\boldsymbol{\theta}}^{n+1} = -\mathbf{K}^{n+1}\boldsymbol{\theta}^{n} + \mathbf{F}^{n+1} \tag{14}$$

$$\boldsymbol{\theta}_{(1)}^{n+1} = \boldsymbol{\theta}^{n} + \Delta t\, \dot{\boldsymbol{\theta}}^{n+1} \tag{15}$$

$$\boldsymbol{\theta}_{(2)}^{n+1} = \boldsymbol{\theta}^{n} + 1/2\Delta t(\dot{\boldsymbol{\theta}}^{n} + \dot{\boldsymbol{\theta}}^{n+1}) \tag{16}$$

where the subscripts in (15) and (16) indicate the order of accuracy of the solution estimates. Since the matrices $\mathbf{K}$ and $\mathbf{F}$ are non-linear (and $\mathbf{M}$ is non-linear when solving the pressure form of Richards equation), it is also necessary to specify the argument they are evaluated with: $\boldsymbol{\theta}_{(1)}^{n+1}$ or $\boldsymbol{\theta}_{(2)}^{n+1}$. In fact, using $\boldsymbol{\theta}_{(2)}^{n+1}$ transforms (14)–(16) into an Adams–Moulton method, while using $\boldsymbol{\theta}_{(1)}^{n+1}$ leads to the backward Euler/Thomas–Gladwell pair as originally designed. Whilst both the Adams–Moulton and Thomas–Gladwell schemes are second-order accurate, we prefer to use $\boldsymbol{\theta}_{(1)}^{n+1}$ to exploit the strong stability properties of the backward Euler scheme and, importantly, to maintain equivalence with existing backward Euler codes.

The parameter set (13) has the following additional advantages [24]:

(1) The forcing vector $\mathbf{F}(\boldsymbol{\theta}, t)$ is evaluated exactly at the end of the time step, making any intermediate interpolations unnecessary;
(2) The first-order solution (15) is the very stable and reliable backward Euler approximation, which is ubiquitous in engineering software and therefore conceptually accessible to practitioners.

The ability to compute two solution estimates of adjacent order of accuracy at the cost of the inversion of a single non-linear system can now be exploited to devise a simple and effective adaptive algorithm.

2.3. Error control and adaptive stepsize variation

The difference between the first- and second-order approximations to the ODEs is a measure of the truncation error of the scheme (15) incurred during the $(n+1)$th time step Δt^{n+1} from $t^n \rightarrow t^{n+1}$:

$$\begin{aligned}\mathbf{e}^{n+1} &= \boldsymbol{\theta}_{(1)}^{n+1} - \boldsymbol{\theta}_{(2)}^{n+1} = \Delta t^{n+1}\dot{\boldsymbol{\theta}}^{n+1} - 1/2\Delta t^{n+1}(\dot{\boldsymbol{\theta}}^{n} + \dot{\boldsymbol{\theta}}^{n+1}) \\ &= 1/2\Delta t^{n+1}(\dot{\boldsymbol{\theta}}^{n+1} - \dot{\boldsymbol{\theta}}^{n}) = 1/2(\Delta t^{n+1})^2\ddot{\boldsymbol{\theta}}^{n}\end{aligned} \tag{17}$$

Equation (17) can be shown through Taylor series expansions to be an $O(\Delta t^2)$ estimate of the local truncation error of the first-order backward Euler approximation.

Having estimated the local error of the time integration at the $(n+1)$th step, a decision can be made whether to accept or reject the time step. Although the infinity norm $\|\mathbf{e}\|_\infty$ could be used to constrain the largest absolute error across the entire soil moisture profile, it is preferable to enforce fractional accuracy by using a relative error norm $\|\mathbf{e}\|_{\mathrm{r}}$. This type of error control has the advantage of making the user tolerance τ independent of the magnitude

Copyright © 2001 John Wiley & Sons, Ltd.

of the solution. The $(n+1)$th time step is accepted if

$$\left\|\mathbf{e}^{n+1}\right\|_{\mathrm{r}} \equiv \max_i \left|\frac{e_i^{n+1}}{\theta_i^{n+1}}\right| \leqslant \tau \tag{18}$$

Regardless of whether the step is accepted or not, the node index of the maximal error is stored as *iCrit*.

The ability of an adaptive time marching method to select the appropriate stepsize for the next time step is paramount for algorithm efficiency. The time step selector is derived by noting that

$$\left\|\mathbf{e}^{n+1}\right\|_{\mathrm{r}} = \max_i \left|\frac{1/2\ddot{\theta}_i^n(\Delta t^{n+1})^2}{\theta_i^{n+1}}\right| = \left|\frac{1/2\ddot{\theta}_{iCrit}^n(\Delta t^{n+1})^2}{\theta_{iCrit}^{n+1}}\right| = \left|\frac{1/2\Delta t^{n+1}(\dot{\theta}_{iCrit}^{n+1} - \dot{\theta}_{iCrit}^n}{\theta_{iCrit}^{n+1}}\right| \tag{19}$$

Enforcing $\|\mathbf{e}^{n+1}\|_{\mathrm{r}} \leqslant \tau$ at t^{n+1} leads to the relation

$$\Delta t^{n+1} \leqslant \sqrt{2\tau\left|\frac{\theta_{iCrit}^{n+1}}{\ddot{\theta}_{iCrit}^n}\right|} \tag{20}$$

Although $\ddot{\boldsymbol{\theta}}^n$ is unknown at t^n, it can be approximated with $O(\Delta t)$ accuracy by $\ddot{\boldsymbol{\theta}}^n \approx \ddot{\boldsymbol{\theta}}^{n-1}$, provided Δt^{n+1} is sufficiently small and $\|\ddot{\boldsymbol{\theta}}(t)\|$ is continuous and bounded in t. Equations (10) and (17) then give $\ddot{\boldsymbol{\theta}}^n \approx \ddot{\boldsymbol{\theta}}^{n-1} = (\dot{\boldsymbol{\theta}}^n - \dot{\boldsymbol{\theta}}^{n-1})/\Delta t^n = 2\mathbf{e}^n/(\Delta t^n)^2$. A first estimate of the $(n+1)$th time step Δt_1^{n+1} is then obtained by substituting this approximation into (20):

$$\Delta t_1^{n+1} \leqslant \Delta t^n \sqrt{\frac{\tau}{\|\mathbf{e}^n\|_{\mathrm{r}}}} = \Delta t^n \sqrt{\tau\left|\frac{\theta_{iCrit}^n}{e_{iCrit}^n}\right|} \tag{21}$$

The adequacy of Δt_1^{n+1} given by (21) is directly verified by (17) and (18) after the solution of (14). If unsatisfactory, the time step is repeated with a smaller stepsize, again predicted by (20), but now substituting the new, more accurate value $\|\mathbf{e}^{n+1}\|_{\mathrm{r}}$ instead of $\|\mathbf{e}^n\|_{\mathrm{r}}$:

$$\Delta t_{j+1}^{n+1} \leqslant \Delta t_j^{n+1} \sqrt{\frac{\tau}{\left\|\mathbf{e}_j^{n+1}\right\|_{\mathrm{r}}}} = \Delta t_j^{n+1} \sqrt{\tau\left|\frac{\theta_{iCrit,j}^{n+1}}{e_{iCrit,j}^{n+1}}\right|} \tag{22}$$

where the subscript j indexes the consecutive time step estimates.

The stepsize selector (20) ensures that time periods where the moisture content varies strongly non-linearly (suggesting large truncation errors in the linear backward Euler approximation) are carefully 'tiptoed' with a reduced stepsize. It is also emphasized that Equations (17)–(22) agree with the generalized formulae developed in ODE analysis to provide an efficient and consistent stepsize variation [23]. Finally, it is possible to implement more general error conditions, which account for both absolute and relative errors. Indeed, in the companion paper [22], we use a mixed absolute-relative error test when applying the error control scheme to the pressure and mixed forms of Richards equation. However, in the case of the moisture-based Richards equation, the solution (the soil moisture) rarely vanishes and therefore the relative error test (18) is sufficient.

In practice, it is prudent to modify the stepsize selection to guard against round-off errors and numerical noise. Safety factor and stepsize multiplier constraints are introduced

Copyright © 2001 John Wiley & Sons, Ltd.

as follows:

$$\Delta t_1^{n+1} = \Delta t^n \times \min\left(s\sqrt{\frac{\tau}{\max(\|\mathbf{e}^n\|_r, \mathrm{EPS})}}, r_{\max}\right) \tag{23}$$

$$\Delta t_{j+1}^{n+1} = \Delta t_j^{n+1} \times \max\left(s\sqrt{\frac{\tau}{\max(\|\mathbf{e}_j^{n+1}\|_r, \mathrm{EPS})}}, r_{\min}\right) \tag{24}$$

Setting $r_{\min} \cong 0.1$ and $r_{\max} \cong 4.0$ smoothes stepsize transitions and guards against dramatic changes in stepsize. The safety factor $s \cong 0.8-0.9$ prevents steps that just fail to meet the local error requirements. EPS is a machine constant ($\sim 10^{-10}$) that prevents floating point errors if $\|\ddot{\boldsymbol{\theta}}^n\|_r \sim 0$.

2.4. Initiation of time stepping

In order to start the time marching from an initial time t_0, the initial derivative $\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t|_{t=t_0} = \dot{\boldsymbol{\theta}}^0$ and an initial time step estimate Δt_0 are necessary. The former can be computed by inverting the governing ODEs:

$$\dot{\boldsymbol{\theta}}^0 = [\mathbf{M}]^{-1}(-\mathbf{K}^0\boldsymbol{\theta}^0 + \mathbf{F}^0) \tag{25}$$

It is noted that, in general, at the $(n+1)$th time step, the derivative vector $\dot{\boldsymbol{\theta}}^n$ is already available from the previous, nth time step. Therefore, Equation (25) is used only at the first time step. Moreover, re-calculating $\dot{\boldsymbol{\theta}}^n$ at each step using the explicit formula (25) makes the scheme equivalent to a Runge–Kutta scheme, which, although still second-order accurate, is only conditionally stable and unsuited to the integration of stiff ODE problems.

If the scheme is applied to semi-discrete PDEs with a non-linear mass matrix $\mathbf{M} = \mathbf{M}(\boldsymbol{\theta})$ (e.g., the pressure form of Richards equation), then Equation (25) still represents a linear system in $\dot{\boldsymbol{\theta}}^0$ (since $\mathbf{M}^0$ is completely determined by the known $\boldsymbol{\theta}^0$). Its inversion cost remains negligible compared to the cost of the non-linear iterations at the subsequent time steps.

An accurate and cheap estimate of the first time step Δt_0 enhances the efficiency of the algorithm and can be obtained as follows:

$$\Delta t_0 = \min\left\{(t_{\text{output}} - t_0), s\sqrt{\tau} \Big/ \max\left(\mathrm{EPS}, \max_i \left|\frac{\dot{\theta}_i^0}{\theta_i^0}\right|\right)\right\} \tag{26}$$

Equation (26) originates in ODE software design and is based on Taylor series analysis [23]. It is consistent with (24) and makes use of all readily available data, including user knowledge (distance to output point t_{output}) and anticipated solution behaviour (given by $\dot{\boldsymbol{\theta}}^0$).

2.5. Local extrapolation

Although the error controller (17) approximates the local truncation error of the first-order accurate backward Euler scheme (15), it is advantageous to store and march forward the higher order Thomas–Gladwell estimate given by (16). This is equivalent to adding the local error estimate at each time step to the lower order approximation itself and is termed 'local extrapolation'. Local extrapolation raises the order of accuracy of the solution and is hence

Copyright © 2001 John Wiley & Sons, Ltd.

widely used in ODE software [23]. It also improves the control of global errors, since (17) is only an approximate and local error estimate.

2.6. Solution of the non-linear systems

Since Richards equation and its semi-discrete analogue (2) are non-linear, implicit integrators give rise to non-linear algebraic systems at each time step. The new automatic time stepping algorithm developed in this paper can be employed with any non-linear solver applicable to (14). In this analysis, the iterative Picard scheme is used because it is simple, does not require Jacobian data and is widely used in practical software for Richards equation. The following linearized system must be inverted at each iteration:

$$[\mathbf{M} + \Delta t\, \mathbf{K}^{n+1,m}]\dot{\boldsymbol{\theta}}^{n+1,m+1} = -\mathbf{K}^{n+1,m}\boldsymbol{\theta}^{n} + \mathbf{F}^{n+1,m} \tag{27}$$

where m is the iteration counter. As discussed above, we employ $\mathbf{K}^{n+1,m} = \mathbf{K}(\boldsymbol{\theta}_{(1)}^{n+1,m})$ and $\mathbf{F}^{n+1,m} = \mathbf{F}(\boldsymbol{\theta}_{(1)}^{n+1,m})$ in order to maintain equivalence with the backward Euler method.

Iterations can be terminated when a relative convergence test of the form $\max_i |(\theta_i^{n+1,m+1} - \theta_i^{n+1,m})/\theta_i^{n+1,m+1}| \leqslant \tau_{\mathrm{PI}}$ (where τ_{PI} is the Picard iteration tolerance) is satisfied. It is sensible to use the same type of error test (enforcing fractional accuracy in the solution) in the truncation error controller and in the non-linear solver. It has been shown that, in general, it may not be necessary or sufficient to set $\tau_{\mathrm{PI}} < \tau$ [28]. Unfortunately, it is difficult to *a priori* establish whether this is the case. In practice we recommend setting $\tau_{\mathrm{PI}} < \tau$ (e.g. $\tau_{\mathrm{PI}} = 0.01\tau$ in this study) to attempt to constrain the residual errors of the non-linear solver below the truncation errors of the time stepping scheme. Previous empirical assessment justifies this approach [25].

2.7. Initial solution estimates for the non-linear iteration

The sensitivity of non-linear iterative processes to the choice of initial solution estimates $\boldsymbol{\theta}^{n+1,0}$ is well-known (e.g. References [20, 29]). Most existing algorithms simply re-use $\boldsymbol{\theta}^{n}$ (e.g. References [4, 9]); some extrapolated methods have also been suggested [30]. In the context of the adaptive scheme, accurate initial estimates can be easily designed, e.g. the quadratic estimate

$$\boldsymbol{\theta}^{n+1,0} = \boldsymbol{\theta}^{n} + \Delta t^{n+1}\dot{\boldsymbol{\theta}}^{n} + 1/2(\Delta t^{n+1})^2\ddot{\boldsymbol{\theta}}^{n-1} \tag{28}$$

The acceleration $\ddot{\boldsymbol{\theta}}^{n-1}$ is given by the finite difference $\ddot{\boldsymbol{\theta}}^{n-1} = (\dot{\boldsymbol{\theta}}^{n} - \dot{\boldsymbol{\theta}}^{n-1})/\Delta t^{n}$. The estimate (28) is clearly more consistent with the $O(\Delta t^2)$ accuracy of the Thomas–Gladwell approximation than $\boldsymbol{\theta}^{n}$ and is easily calculated using information already stored for the adaptive error control algorithm. More importantly, the adaptive time stepping scheme automatically ensures that the initial guesses given by (28) are accurate. The error of the prediction (28) is related to $1/2(\Delta t^{n+1})^2\ddot{\boldsymbol{\theta}}^{n}$ (the truncation error of the backward Euler scheme) and is hence indirectly controlled by the time step selection mechanism.

2.8. Intermediate and final output times

The 'look-ahead' technique is used to meet the intermediate and final output times [23]:

1. Check whether t_{output} can be reached in a single step Δt^{n+1}, i.e.,

$$t_{\mathrm{current}} + \Delta t^{n+1} \geqslant t_{\mathrm{output}}; \tag{29}$$

Copyright © 2001 John Wiley & Sons, Ltd.

2. Yes $\Rightarrow$ truncate Δt to produce output at t_{output}: $\Delta t^{n+1} = t_{\text{output}} - t_{\text{current}}$. Perform time step;
3. No $\Rightarrow$ check whether t_{output} can be reached in two steps Δt^{n+1}, i.e., $t_{\text{current}} + 2\Delta t^{n+1} \geqslant t_{\text{output}}$;
4. Yes $\Rightarrow$ equalise the time steps, that, is set $\Delta t^{n+1} = 1/2(t_{\text{output}} - t_{\text{current}})$. Perform time step;
5. No $\Rightarrow$ proceed with un-altered time step Δt^{n+1}.

The main advantage of this approach is that it avoids undesirable abrupt changes in time step. For the purposes of this study, the simple procedure given above is effective and sufficient.

2.9. Time integration over discontinuities

The proposed error control cannot obtain accurate estimates of the truncation error and time stepsize when discontinuous boundary conditions are imposed, since the solution derivatives become infinite or undefined at the discontinuities. In such cases, the algorithm is stopped at the discontinuity (the look-ahead method (29) can be used to 'land' on any particular time), the initial and boundary conditions adjusted according to the discontinuity and a re-start (25)–(26) performed. This procedure is mathematically proper and efficient, since (i) any numerical approximation relies at least on continuity in the primary variable; (ii) information (e.g. derivatives) prior to the singularity is mathematically irrelevant for the subsequent solution behaviour; and (iii) the marginal cost of internal re-starts is minimal for most reasonable forcing conditions.

The requirement for algorithm re-initialization to handle discontinuous boundary conditions is not a limitation imposed by the adaptive error control scheme. An uncontrolled backward Euler or Thomas–Gladwell time stepping algorithm would incur a large error if simply allowed to integrate over a discontinuity. In particularly severe cases, the non-linear solver may simply not converge. Conversely, the adaptive error control is able to identify a problem and explicitly alert the user. When an automatic internal re-start is implemented, the adaptive scheme integrates over discontinuities with no loss of accuracy and at minimal additional cost (as will be shown in test problem B).

2.10. Incorporation into existing software

An important merit of the proposed adaptive time stepping and error control algorithm is the ease with which it can be incorporated into existing backward Euler codes with fixed or heuristic stepsize selection. Although most backward Euler implementations do not explicitly compute the first derivatives $\mathbf{V}$, these vectors can be back-calculated by differencing the solution:

$$\mathbf{V}^{n+1} = \dot{\boldsymbol{\theta}}^{n+1} = \left(\boldsymbol{\theta}_{(1)}^{n+1} - \boldsymbol{\theta}^{n}\right) / \Delta t^{n+1} \tag{30}$$

The evaluation of (30) and the stepsize selector itself are computationally cheap, especially compared to the inversion of the finite element matrices at each iteration, and should not introduce any noticeable computational burden into the time stepping. However, they will improve the accuracy and efficiency of the codes, making them capable of automatic adaptation to particular solution behaviours. The improved initial estimate for the non-linear iterations, easily available as a by-product of the time step adaptation, can further improve the computational efficiency of the code. We also recommend the look-ahead technique and the internal re-start option to further improve the operational robustness of the code.

Copyright © 2001 John Wiley & Sons, Ltd.

Backward Euler algorithms are widely used in hydrologic and engineering practice and there is a legacy of codes using time step selection rules that are out of step with current ODE integration standards. The practical value of the proposed technique is therefore in the opportunity to upgrade existing software and improve practical computational standards.

3. RESULTS AND DISCUSSION

Two test problems are considered in order to examine the performance of the adaptive scheme and compare it with standard time stepping methods for Richards equation. Problem A represents a monotonic infiltration into an initially very dry soil, driven by time-constant Dirichlet boundary conditions. Whilst a relatively simple problem, it is included to allow direct comparison with other published solutions of Richards equation [4, 15]. Problem B represents a more complex flow regime driven by time-varying boundary conditions. These forcing conditions demand a more dynamic variation in stepsize than problem A.

In both test problems, the domain consists of a 60-cm vertical column of soil with the hydraulic properties described by the van Genuchten constitutive functions:

$$K(\theta) = K_s \theta_e^{1/2} \{1 - (1 - \theta_e^{1/m})^m\}^2 \tag{31}$$

$$D(\theta) = \frac{(1-m)K_s}{\alpha m(\theta_s - \theta_r)} \theta_e^{m-2/2m} \left\{ \frac{1}{(1-\theta_e^{1/m})^m} + (1-\theta_e^{1/m})^m - 2 \right\} \tag{32}$$

where $\theta_e = (\theta - \theta_r)/\theta_s - \theta_r$ and $\theta_s = 0.368$ (saturated moisture content), $\theta_r = 0.102$ (residual moisture content), $\alpha = 0.0335$, $n = 2$, $m = 0.5$ and $K_s = 0.00922\,(\mathrm{cm/s})$. These soil parameters are taken from Celia *et al.* [4] and correspond to a type of New Mexico soil.

In the absence of analytical solutions, surrogate 'exact' solutions are required for the assessment of accuracy and efficiency of the algorithms. The reference solution is approximated numerically by the adaptive scheme with very tight truncation error and iteration tolerances ($\tau = 10^{-8}$ and $\tau_{PI} = 10^{-10}$). This ensures that any differences between 'exact' and 'approximate' solutions are dominated by the truncation error of the 'approximate' solution. Since a relative error test is used in the truncation error controller, the actual errors are also expressed in a relative form, defined as $\varepsilon(t^n) = \max_i |\theta_i^n - \bar{\theta}_i^n / \bar{\theta}_i^n|$, where $\bar{\theta}_i^n$ is the 'exact' solution and i indexes the nodes in the finite element mesh. The error profile is obtained by computing the 'exact' and 'approximate' solutions at a series of *a priori* specified times.

Unless noted otherwise, all solutions are obtained using identical spatial grids comprising 100 linear elements of uniform size. The use of identical spatial approximations isolates the errors introduced by the temporal discretization of the ODE system (2) and facilitates the analysis of temporal accuracy.

In all adaptive runs, the non-linear iteration tolerance is set to $\tau_{PI} = 0.01\tau$ to exclude residual non-linear solver errors, which are superfluous in the temporal error analysis. The auxiliary constraints in (24) are set to $s = 0.85$, $r_{max} = 4.0$ and $r_{min} = 0.1$. The motivation for the choice of these parameters is discussed by Kavetski *et al.* [22], where it is shown that the performance of the adaptive scheme is robust with respect to moderate changes in these parameters.

Copyright © 2001 John Wiley & Sons, Ltd.

The new adaptive time stepping scheme is compared with two currently standard approaches for time integration of Richards equation: uniform and heuristic time stepping schemes.

The stepsize for the uniform scheme was chosen to result in a similar maximum relative error to the presented runs of the adaptive scheme. This allows a direct comparison of the computational effort required by the schemes to obtain solutions of a similar accuracy. In all the results presented for the fixed-step scheme, the standard initial estimates $\boldsymbol{\theta}^{n+1,0} = \boldsymbol{\theta}^n$ for the Picard iterations have been used, instead of the extrapolated approach (28). This allows a direct comparison of the proposed adaptive algorithm with existing implementations.

The heuristic time stepping method is an empirical technique where the time stepsize adjustment for the $(n+1)$th time step is based on the number of iterations (N_{iter}) taken to solve the non-linear system at the nth step:

if $(N_{\text{iter}} < N_{\min}) \rightarrow \Delta t^{n+1} = \Delta t^n \times F_{\text{increase}}$ (if the iteration convergence is 'fast', increase stepsize)
if $(N_{\text{iter}} > N_{\max}) \rightarrow \Delta t^{n+1} = \Delta t^n \times F_{\text{decrease}}$ (if the iteration convergence is 'slow', reduce stepsize)
else $\rightarrow \Delta t^{n+1} = \Delta t^n$ (if the iteration convergence is 'medium', retain stepsize)

Here, $N_{\max/\min}$ and $F_{\text{increase/decrease}}$ are problem-specific empirical constants, either embedded in the code or chosen by the user. This method and similar heuristic schemes are common in Richards equation solvers [9, 15, 19].

The performance parameters for the heuristic scheme are pre-optimized to produce uniform error profiles for a standard flow problem (Test Problem A) and then applied with no adjustment to Test Problem B. This is how similar heuristic schemes are typically used in practice. In order to separate the numerical issues of the order of accuracy and adaptive time step selection, the heuristic stepsize variation is implemented for the Thomas–Gladwell scheme with parameters (13), even though Thomas–Gladwell schemes are not commonly used for Richards equation. This procedure ensures that any difference between the adaptive and heuristic scheme is due solely to the time step selection method. While results are only presented for a heuristic Thomas–Gladwell scheme, a backward Euler scheme with heuristic time stepping has also been implemented and was found to be inferior to the Thomas–Gladwell method.

3.1. Test problem A

This test study consists of modelling monotonic infiltration into the soil column. Boundary moisture contents of $\theta(z=0,t) = 0.2004$ and $\theta(z=60,t) = 0.11$ are imposed. The initial conditions are:

$$\theta(z, t=0) = \begin{cases} 0.11, & z \geqslant 0.6 \\ 0.2004 - \dfrac{0.2004 - 0.11}{0.6} z, & 0 \leqslant z < 0.6 \end{cases} \tag{33}$$

Temporal (and spatial) gradients vary greatly within the highly non-linear region at the toe of the moisture front making the problem a good test of the various time integration schemes.

The solution obtained with the adaptive time stepping scheme is illustrated in Figure 1, which shows the formation of an infiltration (shock) front and its wave-like propagation through the soil column. The figure shows that the solution generated by the adaptive scheme with $\tau = 10^{-2}$ is almost indistinguishable from the exact solution.

Figure 2 shows the error profile of the adaptive time stepping scheme for a range of user-prescribed error tolerances (shown as τ on the figure). The magnitude of the errors is

Copyright © 2001 John Wiley & Sons, Ltd.

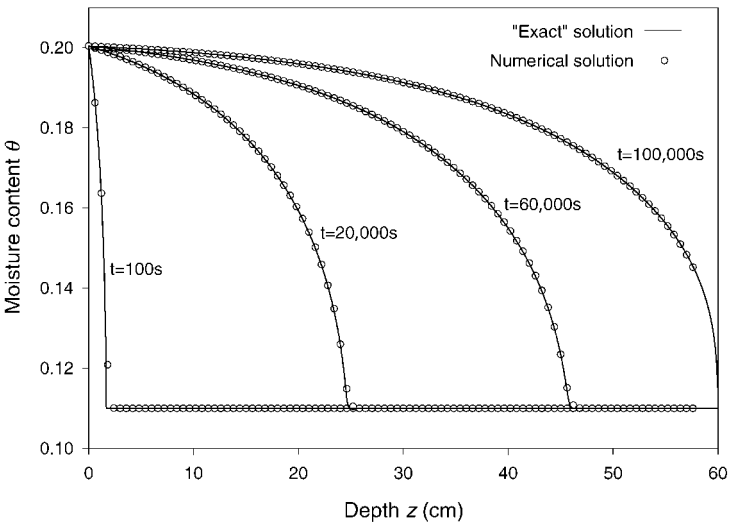

Figure 1. Moisture content as a function of depth at various times throughout the simulation. The solution is obtained with an error tolerance $\tau = 10^{-2}$ and is almost indistinguishable from the 'exact-in-time' solution.

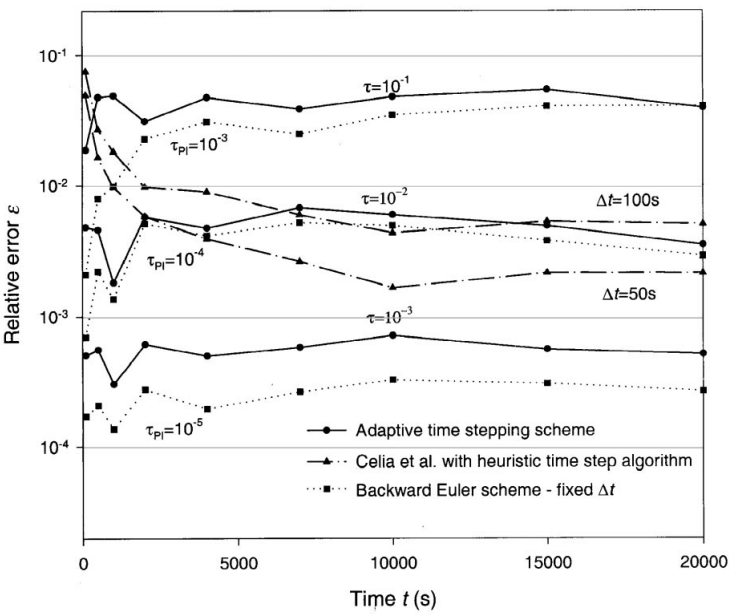

Figure 2. Test problem A: relative errors of: (a) the adaptive scheme for various error tolerances τ; (b) the uniform scheme with various Δt; and (c) the heuristic schemes with non-linear iteration tolerance τ_{PI}. The errors are shown up to $t = 20\,000$ s.

Copyright © 2001 John Wiley & Sons, Ltd.

Table I. Computational cost of the adaptive-stepsize solutions shown in Figure 2. CPU times are for a Pentium II 350 MHz processor.

Truncation error tolerance τ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Non-linear solver tolerance τ_{PI}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Max. relative error in solution	5.58×10^{-2}	7.08×10^{-3}	7.10×10^{-4}	7.34×10^{-5}	7.70×10^{-6}
Number of successful time steps	55	249	788	2472	7782
Number of failed time steps	0	3	1	1	0
Total number of iterations	354	1050	2352	6917	15759
CPU time (s)	0.6	1.5	3.3	9.7	23.6

remarkably uniform throughout the entire integration, attesting to the consistency of the approximation and the reliability of the error control. The scheme has been tested with various spatial discretizations, ranging from 10 to 10 000 elements. In all cases, similar uniform temporal error profiles were obtained, confirming the robustness of the adaptive time stepping mechanism with respect to the spatial discretization.

The second order convergence of the Thomas–Gladwell estimates is evident from Table I, with the number of time steps increasing by a factor of $\approx\sqrt{10}$ for a reduction in tolerance (and actual errors) by a factor of ≈ 10. This trend confirms that the algorithm remains second-order convergent under variable-stepsize conditions.

The error profile obtained with the adaptive scheme is compared to those of the uniform and heuristic schemes in Figure 2. The figure shows that the backward Euler scheme with uniform time step sizes suffers from significant errors in the initial flows, with accuracy improving with time. If accurate intermediate results are needed for the early times of the simulation, the backward Euler approximation with fixed step size is inadequate unless very small time steps are used. In addition, the first-order convergence of the scheme becomes a noticeable limitation as the accuracy requirements are increased.

The heuristic scheme produces approximations that are qualitatively similar to those of the adaptive scheme, with more or less uniform temporal errors throughout the integration (Figure 2). However, for this problem the heuristic parameters have been optimized in a trial and error process, with *a priori* knowledge of the exact solution.

The relative algorithm efficiency can be assessed on the basis of actual accuracy for a given computational cost, (Tables I–III). The total number of iterations can be used as the measure of computational effort since the CPU time is governed by the total number of matrix inversions, rather than by the number of time steps. A single (Picard) iteration of the adaptive scheme is identical to the iterations of the uniform and heuristic stepsize implementations. Only after the non-linear solver converges does the marginal computational overhead associated with the error estimation and stepsize selection take place. It is clear from Tables I and II that the automatic time stepping algorithm is more accurate than the uniform backward Euler scheme for a similar number of time steps and iterations (and time steps). The efficiency of the heuristic scheme (Table III)is roughly equivalent to the adaptive scheme. This can be expected from the results in Figure 2, since the actual approximation formulae are identical and the algorithms differ only in stepsize selection—similar error profiles imply similar time step histories and hence similar efficiency.

The stepsize evolution produced by the adaptive scheme is quite intuitive (Figure 3). Early times ($t < \sim 4000$ s) are characterized by very rapid and highly non-linear moisture flows

Copyright © 2001 John Wiley & Sons, Ltd.

Table II. Computational cost of the uniform-stepsize backward Euler solutions shown in Figure 2.

Time step size Δt (s)	100	50
Non-linear solver tolerance τ_{PI}	10^{-3}	10^{-4}
Max. relative error in solution	7.49×10^{-2}	4.92×10^{-2}
Number of time steps	1000	2000
Number of non-linear iterations	2176	4992
CPU time (s)	3.1	7.1

Table III. Time stepping parameters and computational cost of the Thomas–Gladwell solutions with optimized heuristic stepsize selection (Figure 2). Initial time step $\Delta t_0 = 1$ s in all cases. Note that the improved initial estimate (28) has been used by the Picard non-linear solver.

Non-linear solver tolerance τ_{PI}	10^{-3}	10^{-4}	10^{-5}
$N_{\min}$	8	4	3
$N_{\max}$	15	10	6
F_{increase}	1.1	1.05	1.02
F_{decrease}	0.95	0.95	0.95
Max. relative error in solution	1.44×10^{-1}	5.20×10^{-3}	3.22×10^{-4}
Number of time steps	103	352	1213
Number of non-linear iterations	437	1218	3347
CPU time (s)	0.6	1.7	4.9

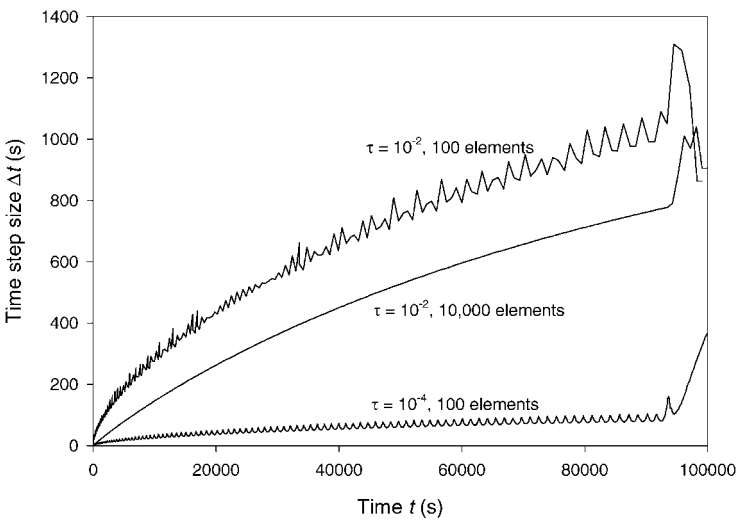

Figure 3. Time step variation of the adaptive scheme throughout the integration of test problem A.

due to abrupt forcing. Fine time steps ($\Delta t \sim 0.3$–3 s) are selected in this time period. As the infiltration front reaches its constant velocity and shape during the remainder of the simulation, the initially very fast stepsize growth decelerates and Δt approaches a constant value. Finally, a dramatic rise in stepsize takes place as the infiltration front reaches the end of the soil column and the system equilibrates.

Copyright © 2001 John Wiley & Sons, Ltd.

The stepsize variation is also related to the spatial approximation. It is observed that time step oscillations occur when a 100-element spatial grid is used. These oscillations disappear for a 10 000-element grid (which effectively generates a spatially continuous ODE system). The oscillations are associated with the abrupt transition of the moisture front across the boundaries of the elements and are probably exacerbated by the numerical discontinuity of the moisture fluxes at the nodes of the linear elements ($\partial\theta/\partial z$ is undefined at the nodes). This observation reveals a useful feature of mathematically based time stepping algorithms—they not only directly control time errors, but also indirectly assess the adequacy of the spatial approximation. It is seen from Figure 3 that the minimization of spatial errors produced by the dense grid leads to a smooth variation in time step size that is qualitatively consistent with the dissipative character of the solution.

The time step profile in Figure 3 also explains the poor performance of the uniform step size algorithm. In order to achieve a given error, the uniform time stepping algorithm must choose the smallest time step required throughout the simulation. In this case, the time step size is determined by the large errors caused by the strong non-linearities at the beginning of the simulation. The errors decrease at later times as these non-linearities weaken. As the overall error in the solution is dominated by the initial errors, the effort expended in using small time steps is wasted for most of the simulation.

3.2. Test problem B

The second test study alters the forcing condition at the top of the soil column, imposing a time dependent discontinuous Dirichlet boundary condition of the following form:

$$\theta(z=0,t)=\begin{cases} 0.15+0.03\ \sin(16\,500-0.00015t) & 0<t\leqslant 50\,000 \\ 0.25 & 50\,000<t\leqslant 65\,000 \\ 0.14 & t>65\,000 \end{cases} \tag{34}$$

The moisture levels described by (34) follow a smooth sinusoid (one full period) and then a sharp square pulse occurs at $t=50\,000$ s, followed by a return to low saturation levels at $t=65\,000$ s. The boundary condition hence involves both smooth continuous variations as well as sharp discontinuous changes. The flow patterns under this forcing are more complex than in problem A, and involve a propagation of oscillating waves down the soil column, both upward and downward flows, followed by a sharp shock when the step discontinuity at $t=50\,000$ s is reached. In order to maintain accuracy and efficiency, a time stepping algorithm must be able to adjust stepsize much more dynamically than was required for the monotonic imbibition in the previous test problem.

The error profiles for the adaptive, heuristic and uniform stepsize solutions with the boundary condition (34) are shown in Figure 4, with the run time statistics in Tables IV–VI. The performance settings of the heuristic scheme are retained from the previous case, where they led to an accurate and cost-effective solution. The uniform stepsize in the backward Euler approximation was adjusted to lead to a maximum error of approximately 1%. No adjustment of the adaptive scheme is necessary.

Figure 4 shows that the more complex boundary conditions did not significantly affect the ability of the adaptive scheme to achieve uniform temporal error profiles. At no time within the simulation, neither during the smooth continuous variation, nor after the sharp discontinuous jump in the boundary conditions, did the time approximation errors exceed the

Copyright © 2001 John Wiley & Sons, Ltd.

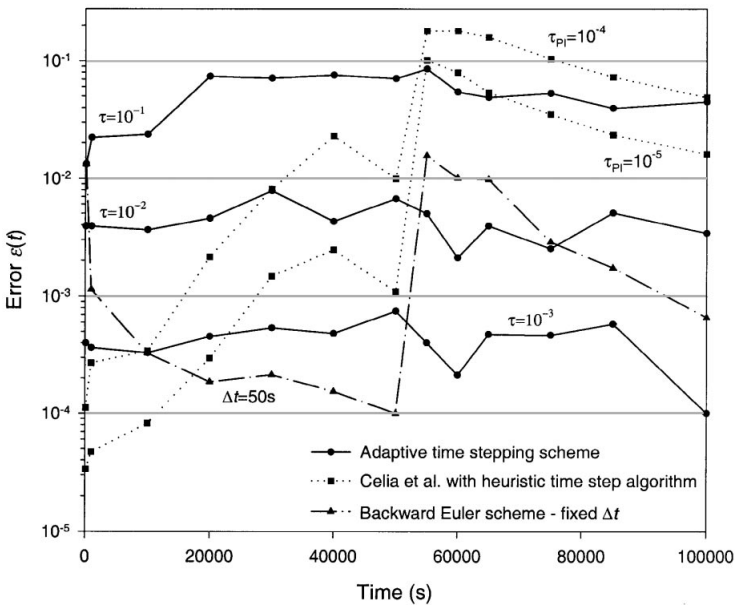

Figure 4. Test problem B: relative errors of the adaptive, uniform and heuristic schemes.

Table IV. Computational cost of the adaptive-stepsize solutions shown in Figure 4. CPU times are for a Pentium II 350 MHz processor.

Truncation error tolerance τ	10^{-1}	10^{-2}	10^{-3}	10^{-4}
Non-linear solver tolerance τ_{PI}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Max. relative error in solution	8.56×10^{-2}	7.83×10^{-3}	7.49×10^{-4}	7.26×10^{-5}
Number of successful time steps	82	335	1052	3291
Number of failed time steps	1	15	6	15
Total number of iterations	501	1405	3150	88
CPU time (s)	0.7	1.9	4.3	12.3

Table V. Time stepping parameters and computational cost of the Thomas–Gladwell solutions with optimized heuristic stepsize selection (Figure 4). Initial time step $\Delta t_0 = 1$ s in all cases.

Non-linear solver tolerance τ_{PI}	10^{-4}	10^{-5}
$N_{\min}$	8	4
$N_{\max}$	15	10
F_{increase}	1.1	1.05
F_{decrease}	0.95	0.95
Max. relative error in solution	1.80×10^{-1}	1.01×10^{-1}
Number of time steps	251	767
Number of non-linear iterations	970	2460
CPU time (s)	1.3	3.3

Copyright © 2001 John Wiley & Sons, Ltd.

Table VI. Computational cost of the uniform-stepsize backward Euler solutions shown in Figure 4.

Time step size Δt (s)	50
Non-linear solver tolerance τ_{PI}	10^{-4}
Max. relative error in solution	1.4×10^{-2}
Number of time steps	2000
Number of non-linear iterations	4204
CPU time (s)	4.0

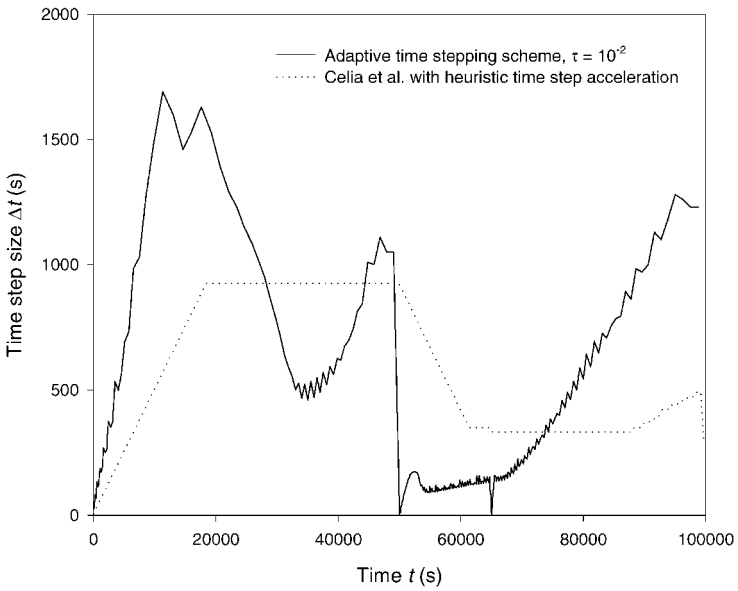

Figure 5. Time step variation of the adaptive scheme throughout the integration of test problem B.

user-prescribed tolerance. This finding demonstrates the reliability of the proposed scheme. Conversely, the uniform and heuristic schemes produce approximations with very uneven distributions of errors. For example, the uniform backward Euler scheme is very accurate at $20\,000 < t < 50\,000$ s, but, after the jump in the boundary condition, the errors increase by a factor of 100. It is also noted that the cost of this uniform solution (2000 time steps, 4200 iterations) is the largest of all simulations shown in Figure 4. The heuristic scheme also has problems adjusting to the solution behaviour. Its error grows progressively throughout the simulation, even during the smooth sinusoidal variations at the Dirichlet node. Similarly to the case with the uniform scheme, the jump discontinuity at $t = 50\,000$ s leads to a dramatic increase in error (also by about 2.5 orders of magnitude) in the heuristic scheme.

The time step histories for the adaptive and heuristic schemes are shown in Figure 5. It is immediately apparent that the approximation of flows induced by the non-trivial time dependent Dirichlet boundary condition requires a time step variation that is subtler than that required for the monotonic imbibition in the first test problem. The peaks in stepsize

Copyright © 2001 John Wiley & Sons, Ltd.

correspond to the time where the sinusoidal moisture variation at the Dirichlet boundary reaches its low point, and to the period following the decrease of the boundary saturation after the square wave recedes. The proposed adaptive algorithm extracts the relevant error information from internal approximations to the time derivatives and, using flexible stepsize multipliers, successfully adapts to the solution behaviour (as seen in Figure 4).

Conversely, the performance of the non-linear (Picard) solver is insufficiently informative about the likely temporal errors, and so the heuristic scheme is unable to adapt neither to the smooth continuous variation in forcing at $t < 50\,000$ s, nor to the subsequent square pulse. In addition, it can be seen that the fixed stepsize multipliers excessively constrain the heuristic scheme. The fact that these multipliers performed very well for the previous test problem does not translate into good performance when the forcing patterns are altered. Finally, the user cannot explicitly control the errors when employing the heuristic scheme because the non-linear solver tolerances (10^{-4} and 10^{-5}) are not directly related to the temporal errors in the solution (1.8×10^{-1} and 1.0×10^{-1}).

3.3. General remarks on heuristic vs adaptive time stepping

Although the heuristic stepsize schemes can be optimized to produce error profiles that are qualitatively similar to those of the adaptive scheme, they have the following limitations:

(1) The optimal heuristic time stepping parameters depend on the particular flow regime in the simulation. In the present case, a trial and error optimization process was employed, with *a priori* knowledge of the exact solution. In practice, where the solution behaviour is largely unknown, the accuracy and efficiency of the integration becomes dependent on the ability of the user to correctly determine the heuristic time stepping parameters. The primary means of specifying the accuracy of the heuristic scheme is the non-linear iteration tolerance. This tolerance is not directly related to the temporal accuracy of the solution;
(2) Variation of the empirical factors required during the simulation of non-trivial flow conditions (e.g. with intermittent rain, drainage or recharge) may be necessary, but codes implementing the heuristic scheme usually employ a single set of heuristic parameters. It is noted that while the adaptive scheme also introduces what can be viewed as performance-tuning parameters (Equations (23) and (24)), these play only an auxiliary, safeguarding role to the main time step selection mechanism and need not be changed from simulation to simulation. Conversely, the heuristic parameters constitute the very essence of the heuristic scheme and, ideally, must be optimized and verified by the user for each particular flow problem;
(3) The stepsize is linked to the non-linear solver via an empirical relation that depends on the particular non-linear solver used. If a different, e.g. more efficient scheme is implemented, or more accurate initial estimates for the iterations are supplied, the empirical factors must be changed. Moreover, heuristic rules based on the number of iterations per time step are fundamentally incompatible with non-iterative non-linear solvers. Paniconi *et al.* [9] describe some non-iterative solvers that hold promise for Richards equation, but common heuristic schemes cannot exploit their advantages. Conversely, virtually any non-linear solver can be used to invert the non-linear implicit systems of the adaptive time stepping scheme. The Picard iteration scheme is chosen here to simplify the presentation, although any other approach is also easily implemented. It is much simpler to upgrade the non-linear solver within the adaptive scheme than within a heuristic algorithm.

Copyright © 2001 John Wiley & Sons, Ltd.

3.4. *Verification of results in a practical simulation* (*with unknown solution*)

In current practice, the user of a fixed-stepsize code arrives to the final choice of stepsize only after convergence studies with a series of uniform time steps, and strongly non-linear local behaviour may significantly delay convergence across the entire temporal grid. The verification of the heuristic scheme may include altering the heuristic parameters, the quantitative effect of which is not immediately apparent for a particular problem. Conversely, the validation of the automatic algorithm is simpler, requiring the re-running the scheme with a range of error tolerances (e.g. $\tau = 10^{-2} \rightarrow 10^{-3} \rightarrow 10^{-4}$), which should constrain the temporal error profile more or less uniformly. Naturally, if the solutions do not converge as the tolerance is reduced, the results are suspect. A somewhat simpler verification methodology is a valuable practical advantage in addition to the automatic stepsize optimization, making the adaptive scheme more reliable and cheaper in terms of the ultimate CPU and user effort per simulation than either the uniform or heuristic algorithms.

3.5. *General applicability of the proposed algorithm*

Finally, although the present analysis focuses on the unsaturated flow equation, non-linear PDEs with first-order time derivatives arise in geomechanical, hydrological and geothermal studies, as well as in other areas of engineering and science. In many cases, strong non-linearity, stability or implementation considerations favour a simple and robust low-order method, or there may be a legacy of backward Euler codes in current use. The proposed adaptive error control and stepsize variation methodology hence presents an attractive opportunity to improve applied numerical methods for the solution of PDEs in other engineering and scientific applications.

4. CONCLUSIONS

A practical approach for improving the mathematical accuracy and computational efficiency of temporal numerical simulations of unsaturated flows has been presented. Although typical time stepping algorithms for Richards equation disregard derivative data from previous time steps, it is advantageous to make use of this information. Existing and new software based on the backward Euler scheme can be equipped with simple and inexpensive error control and adaptive stepsize selection. Second-order temporal accuracy and improved initial estimates for the non-linear solver are then obtained at minimal additional cost. The resulting algorithm is robust, fully automated and capable of accurate and efficient solution of the highly non-linear Richards equation in the context of an engineering application. The simplicity and generality of the proposed methodology make it accessible to practitioners in hydrogeology, water resources and other areas of environmental engineering and applied science.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers for their insightful comments and constructive criticisms.

Copyright © 2001 John Wiley & Sons, Ltd.

REFERENCES

1. Philip JR. Theory of Infiltration. *Advances in Hydroscience* 1965; **5**:215–296.
2. Huyakorn PS, Pinder GF. *Computational Methods in Subsurface Flow*. Academic Press: San Diego, 1985.
3. Gray WG, Hassanizadeh S. Paradoxes and realities in unsaturated flow theory. *Water Resources Research* 1991; **27**(8):1847–1854.
4. Celia MA, Bouloutas ET, Zarba RL. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research* 1990; **26**(7):1483–1496.
5. Tocci MD, Kelly CT, Miller CT. Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines. *Advances in Water Resources* 1997; **20**(1):1–14.
6. Bergamaschi L, Putti M. Mixed finite elements and Newton-type linearizations for the solution of Richards' equation. *International Journal for Numerical Methods in Engineering* 1999; **45**: 1025–1046.
7. Ju SH, Kung KJS. Mass types, element orders and solution schemes for the Richards equation. *Computers and Geosciences* 1997; **23**(2):175–187.
8. Abriola LM, Lang JR. Self-adaptive hierarchic finite element solution of the one-dimensional unsaturated flow equation. *International Journal for Numerical Methods in Fluids* 1990; **10**(3):227–246.
9. Paniconi C, Aldama AA, Wood EF. Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation. *Water Resources Research* 1991; **27**(6):1147–1163.
10. Paniconi C, Putti M. A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems. *Water Resources Research* 1994; **30**(12):3357–3374.
11. Fassino C, Manzini G. Fast-secant algorithms for the non-linear Richards equation. *Communications in Numerical Methods in Engineering* 1998; **14**(10):921–930.
12. Jones JE, Woodward CS. Preconditioning Newton–Krylov methods for variably saturated flow. *Proceedings of the XIII International Conference on Computational Methods in Water Resources*, Calgary, Canada, Balkema: Rotterdam, 2000.
13. Kelley CT, Miller CT, Tocci MD. Termination of Newton/Chord iterations and the method of lines. *SIAM Journal of Scientific Computing* 1998; **19**(1):280–290.
14. Miller CT, Williams GA, Kelley CT, Tocci MD. Robust solution of Richards' equation for nonuniform porous media. *Water Resources Research* 1998; **34**(10):2599–2610.
15. Rathfelder K, Abriola LM. Mass conservative numerical solutions of the head-based Richards equation. *Water Resources Research* 1994; **30**(9):2579–2586.
16. Wood WL. *Practical Time Stepping Schemes*. Oxford University Press: Oxford, 1990.
17. Babajimopoulos C. A Douglas–Jones predictor–corrector program for simulating one-dimensional unsaturated flow in soil. *Groundwater* 1991; **29**(2):267–270.
18. Baker DL. Applying higher order DIRK time steps to the 'modified Picard' method. *Groundwater* 1995; **33**(2):259–263.
19. Celia MA, Binning P. A mass conservative numerical solution for two-phase flow in porous media with application to unsaturated flow. *Water Resources Research* 1992; **28**(10):2819–2828.
20. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in Fortran-77*: *The Art of Scientific Computing* (2nd edn). Cambridge University Press: New York, 1992.
21. Williams GA, Miller CT. An evaluation of temporally adaptive transformation approaches for solving Richards' equation. *Advances in Water Resources* 1999; **22**(8):831–840.
22. Kavetski D, Binning P, Sloan SW. Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation. *Advances in Water Resources* 2001; **24**(6):595–605.
23. Shampine LF. *Numerical Solution of Ordinary Differential Equations*. Chapman and Hall; New York, 1994.
24. Sloan SW, Abbo AJ. Biot consolidation analysis with automatic time stepping and error control. Part 1: Theory and implementation. *International Journal for Numerical and Analytical Methods in Geomechanics* 1999; **23**:467–492.
25. Sloan SW, Abbo AJ. Biot consolidation analysis with automatic time stepping and error control. Part 2: Applications. *International Journal for Numerical and Analytical Methods in Geomechanics* 1999; **23**:493–529.
26. Ruge P. A priori local error estimation with adaptive time-stepping. *Communications in Numerical Methods in Engineering* 1999; **15**(7):479–491.
27. Thomas RM, Gladwell I. Variable-order variable-step algorithms for second-order systems. Part 1: The methods. *International Journal for Numerical Methods in Engineering* 1988; **26**:39–53.
28. Nørsett SP, Thomsen PG. Local error control in SDIRK-Methods. *BIT* 1986; **26**:100–113.
29. Ortega JM, Rheinboldt WC. *Iterative Solution of Nonlinear Equations in Several Variables*. San Diego, CA: Academic Press, 1970.
30. Cooley RL. Some new procedures for numerical solution of variably saturated flow problems. *Water Resources Research* 1983; **19**(5):1271–1285.

Copyright © 2001 John Wiley & Sons, Ltd.