INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2006; **65**:1088–1112
Published online 19 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1481

A non-iterative algorithm based on Richardson's extrapolation. Application to groundwater flow modelling

Fernando G. Basombrío[1,*,†], Luis Guarracino[2] and Marcelo J. Vénere[3]

[1]*Instituto Balseiro, Universidad Nacional de Cuyo—CNEA, 8400 Bariloche, Argentina*
[2]*Facultad de Cs. Astronómicas y Geofísicas, Univ. Nac. de La Plata, Argentina*
[3]*PLADEMA, CNEA and Univ. Nac. del Centro de la Prov. de Bs. As., Tandil, Argentina*

SUMMARY

In this report, a promising third-order algorithm based on Richardson's extrapolation and Crank–Nicholson's method is studied. The algorithm, having a wide scope for applications, was initially intended to be used for time integration on non-linear advection–diffusion problems. Then stability, low oscillations and low additional diffusion become important numerical properties. It is shown that this method, using three single steps for extrapolation, is very close to be unconditionally stable. Under the previous requirements it behaves essentially as well as Crank–Nicholson's scheme but with a better performance, at least in relation to spurious oscillations, and perhaps not so good for dispersion properties. In addition the method is third-order accurate, and provides useful information related to time integration error to be used for time step control. These features indicate that the algorithm is a good candidate to efficiently solve problems with significant variations in time. Additionally, its non-iterative version saves important amounts of CPU time, specially for applied 3D modelling. To illustrate the good performance of the algorithm we include several tests comparing it with Crank–Nicholson's scheme, and a 2D application to variably saturated groundwater flow modelling. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: efficient transient scheme; third-order accurate scheme; Richardson's extrapolation; algorithms testing

1. INTRODUCTION

The advection–diffusion equation is one for which numerical solution procedures continue to exhibit significant limitations for certain problems of physical interest. This is specially true for

*Correspondence to: F. G. Basombrío, Instituto Balseiro, Universidad Nacional de Cuyo—CNEA, 8400 Bariloche, Argentina.
†E-mail: fgbe@yahoo.com.ar

Contract/grant sponsor: ANPcyT; contract/grant number: PICT 12-03239

Received 26 July 2005
Accepted 1 August 2005

large three-dimensional cases with different scale features, such as the transport of contaminants in the presence of multiple pumping activities. In these cases the analyst must deal not only with the natural difficulties of defining and meshing a three-dimensional domain with difference of two or three order of magnitude in the element size, but also with the selection of a time step compatible with the smaller elements. To reduce the work involved in mesh generation, the use of adaptive techniques based on meshes with simplex elements (triangles in 2D or tetrahedral in 3D) is highly recommended (see References [1, 2]). In the present work we choose first-order simplex elements because of their geometric flexibility. Higher order elements could be used, but for complex geometry or problems with singularities, where very small elements are needed, they may become computationally expensive. Additionally these methods give information of the current discretization error, allowing the user to know when the mesh is adequate or when it is not.

It is well known that if conditionally stable methods are used for time integration, the time step must be compatible with the smaller element size to avoid instabilities. This is a drawback to the use of adaptive techniques because the necessary time step will lead to an unaffordable computational cost. In this paper we study a third-order scheme based on the classical Crank–Nicholson's (or trapezoidal rule) method together with Richardson's extrapolation, initially proposed by Módena *et al.* [3]. Richardson's extrapolation give us information on the discretization error of time integration, allowing an automatic adaptation of time step. The combination of this method with adaptive techniques for the space discretization provides a fully adaptive scheme that has a great impact on the computational cost and additionally let the user know and control all discretization errors involved in the analysis. We also found that time step adaptation is a relevant tool to control convergence when non-linear coefficients are present.

Our main purpose is to analyse general properties of the above-mentioned Crank–Nicholson algorithm combined with Richardson extrapolation (see also Reference [4]), and to illustrate the performance of the method via tests, numerical experiments and an application to groundwater flow modelling. It is today well-known [5–7] that the trapezoidal rule possesses remarkable dynamical properties, i.e. amongst all the consistent and strictly zero-stable linear multistep and Runge–Kutta schemes, the trapezoidal rule is the unique *strictly regular* method. That means, a method which (1) exactly reproduces the equilibria (fixed points) of the original system of ordinary differential equations (ODE) without adding spurious ones; (2) for any positive value of the time step, does not admit real period-2 solutions; (3) shows Hopf bifurcations, if and only if the original ODE present them, for exactly the same values of the parameter. Moreover, it is unconditionally stable and order $\mathcal{O}(\Delta t^2)$ accurate. Although some practical drawbacks such as spurious time oscillations are present, the method appears to be a good choice. We shall see that the proposed extrapolated algorithm at least tends to correct part of this deficiency, being additionally third-order accurate. Unfortunately, for the new algorithm some of the above-mentioned properties of Crank–Nicholson's scheme are partially lost, and in some cases we observed that it behaves as more dispersive. The analysis will be mainly focused to a non-iterative algorithm (the linearized trapezoidal rule), owing to the relevance of such kind of schemes in 3D modelling.

In Section 2 we describe the extrapolated algorithm, discussing in Section 3 its general properties. Section 4 contains an application to variably saturated groundwater flow modelling in a 2D domain. Finally, in Section 5 the main conclusions are collected.

2. THE RICHARDSON'S EXTRAPOLATED ALGORITHM

We deal with time discretization of an autonomous system of m ODE for the m unknown functions $U^{\mathrm{T}}(t) = [U_1(t), \ldots, U_m(t)]$,

$$\dot{U} = F(U) \tag{1}$$

Typically, this system comes from space discretization of partial differential equations.

Let Δt be the time step, $r > 1$ the number of steps of Richardson's method and $q = 1/r < 1$. Starting from the conventional initial condition U_n, the method consists of the following steps (see for example Reference [8]):

(i) Evaluate the solution in one time step Δt:

$$U_{n+1}^{(\Delta t)} = U_n + \Delta t[(1-\alpha)F(U_n) + \alpha F(U_{n+1}^{(\Delta t)})], \quad \alpha \in [0, 1] \tag{2}$$

In principle (because $U_{n+1}^{(\Delta t)}$ is an implicit unknown of the previous non-linear equation; see later), this expression defines the mapping,

$$U_{n+1}^{(\Delta t)} = f^{(\Delta t)}(U_n) \tag{3}$$

(ii) Evaluate the iterated solution for r time steps $q\Delta t$:

Now we have,

$$U_{n+1}^{(q\Delta t)} = U_{n+1-q}^{(q\Delta t)} + q\Delta t[(1-\alpha)F(U_{n+1-q}^{(q\Delta t)}) + \alpha F(U_{n+1}^{(q\Delta t)})] \tag{4}$$

defining,

$$U_{n+1}^{(q\Delta t)} = f^{(q\Delta t)}(U_{n+1-q}^{(q\Delta t)}) \tag{5}$$

or,

$$U_{n+1}^{(q\Delta t)} := [f^{(q\Delta t)} \circ \cdots \circ f^{(q\Delta t)}](U_n) \quad (r \text{ times}) \tag{6}$$

(iii) Proceed with Richardson's extrapolation (now it is convenient to write $U_n \equiv U_n^{\mathrm{RE}}$):

$$U_{n+1}^{\mathrm{RE}} = f^{\mathrm{RE}}(U_n^{\mathrm{RE}}) := (1-q^p)^{-1}\{f^{(q\Delta t)} \circ \cdots \circ f^{(q\Delta t)} - q^p f^{(\Delta t)}\}(U_n^{\mathrm{RE}}) \quad (r \text{ times}) \tag{7}$$

where the superscript 'RE' means 'Richardson's Extrapolation'. More specifically, this notation is kept here for the 'Richardson's Extrapolated' trapezoidal rule. The r-iterated of $f^{(q\Delta t)}$ appears in brackets (6). Here p is the order of schemes (2) and (4), that is $p = 1$ if $\alpha \neq \frac{1}{2}$ and $p = 2$ if $\alpha = \frac{1}{2}$. For the case $\alpha = \frac{1}{2}$ the algorithm (7) becomes third-order accurate [8]. The abbreviation TR will be used for 'Trapezoidal Rule'. The method proposed in Reference [3] uses $\alpha = \frac{1}{2}$ and $r = 3$.

Remark

Similar considerations can be made for the 'internal α-method':

$$U_{n+1} = U_n + \Delta t\, F((1-\alpha)U_n + \alpha U_{n+1}) \tag{8}$$

If the original problem (1) is non-linear, methods (2) and (4) are also non-linear and in consequence, they may not have a unique definition. Then we adopt an alternate linearized

algorithm with the same order, avoiding such difficulty as well as iterations and other difficulties discussed in the next section. It is worth pointing out that non-iterative algorithms become mandatory for 3D modelling. This method is obtained by linearizing the α-method and later performing Richardson's extrapolation.

Newton–Raphson's iterations for the α-method are written as

$$U_{n+1}^{(k)} = U_{n+1}^{(k-1)} - [D\varphi(U_{n+1}^{(k-1)})]^{-1}\varphi(U_{n+1}^{(k-1)}), \quad k = 1, 2, \ldots, \quad U_{n+1}^{0} = U_n \tag{9}$$

where the superscript (k) denotes iteration level and,

$$\varphi(U_{n+1}) := U_{n+1} - U_n - \Delta t[(1-\alpha)F(U_n) + \alpha F(U_{n+1})] \tag{10}$$

is the residual and

$$D\varphi = \mathbf{1} - \alpha\Delta t \mathrm{DF}(U_{n+1}) \tag{11}$$

is the tangent matrix.

The linearized α-algorithm is simply obtained by performing just one Newton's iteration. The (implicit) expression is then,

$$U_{n+1} = U_n + \Delta t[F(U_n) + \alpha \mathrm{DF}(U_n) \cdot (U_{n+1} - U_n)] \tag{12}$$

As expected, this is exactly the same as for the 'internal' α-method. It is not difficult to see that the step $U_n \to U_{n+1}$ is correctly defined, if and only if $1/\alpha\Delta t$ is not a spectral point of the matrix $\mathrm{DF}(U_n)$.

Remark

By 'linearized algorithm' we mean that the unknown U_{n+1} appears linearly in the defining equation. But by no means the mapping $U_n \to U_{n+1}$ is linear.

The linearized scheme consists of using the linearized version (12) to define (3) and (5), instead of (2) and (4). It can be easily shown that if $\alpha = \frac{1}{2}$ scheme (12) is also $\mathcal{O}(\Delta t^2)$, and thus the final Richardson's extrapolated algorithm, obtained with (7) and (12), becomes $\mathcal{O}(\Delta t^3)$. This important scheme is designated by REL and it will be analysed in next sections.

3. MAIN PROPERTIES OF RICHARDSON'S EXTRAPOLATED ALGORITHM

Now we discuss various properties of RE and REL time discretization approaches. We wish to compare the ability of both schemes to correctly reproduce the dynamics of the original continuous problem (1).

3.1. Local stability

Both (non-linear and linearized) versions of Crank–Nicholson's method are unconditionally stable in a neighbourhood of any stable fixed point U^* of (2) and (12) (with $\alpha = \frac{1}{2}$). That means, the scheme preserves the stability of U^* as equilibrium of (1) [9] (local stability). The *region of absolute stability* $\mathcal{S}$ [7], i.e. the set of points $z = \lambda\Delta t \in \mathbb{C}$ such that the amplification factor $g(z)$ lies in the unit circle, is the left half complex plane ($\lambda \in \mathbb{C}$ is the constant of the

linear scalar test equation $\dot{u} = \lambda u$). Normally, λ is taken from the spectrum $\{\lambda_i\}$, $i = 1, \ldots, m$ of the gradient matrix $\mathrm{DF}(U^*)$. The dimension of this spectrum is an inverse of time. In what follows it will be convenient to introduce the *dimensionless spectrum* $\{\lambda_i \Delta t\} \equiv \{z_i\}$.

The *linear stability function* of the α-scheme, obtained by linearizing (2) around the fixed point U^* (amplification factor as a function of z, belonging to the dimensionless spectrum of $\mathrm{DF}(U^*)$) is given by

$$g_\alpha(z) \equiv \frac{1 + (1-\alpha)z}{1 - \alpha z} \tag{13}$$

Consequently, for the Richardson's extrapolated scheme (7) we get (using the chain rule and linearizing again),

$$g_{\mathrm{RE}}(z) = \frac{[g_\alpha(qz)]^r - q^p g_\alpha(z)}{1 - q^p} \tag{14}$$

Remark
For the linear case, RE and REL schemes are obviously indistinguishable. Thus, $g_{\mathrm{RE}} \equiv g_{\mathrm{REL}}$.

Unfortunately, with only two sub-steps for Richardson's extrapolation the important unconditional stability property of the TR is lost for RE and REL methods. But this property is essentially recovered if one uses instead $r = 3$ (and odd integers excepting 1; see next paragraph). Of course, this will increase the computational cost because now, for each time step, four resolutions of the basic problem (2) or (12) (with $\alpha = \frac{1}{2}$) will be needed. It can be shown that for three sub-steps, $\mathscr{S}$ fills the whole left half plane with the only exception of a thin layer along the imaginary axis [10]. Thus, RE scheme is *practically* unconditionally stable. For consistency reasons, such layer shrinks to zero just over the real axis (see Reference [10, Figure 1]). The scheme is in fact *unconditionally stable for real spectra*. In Figure 1,

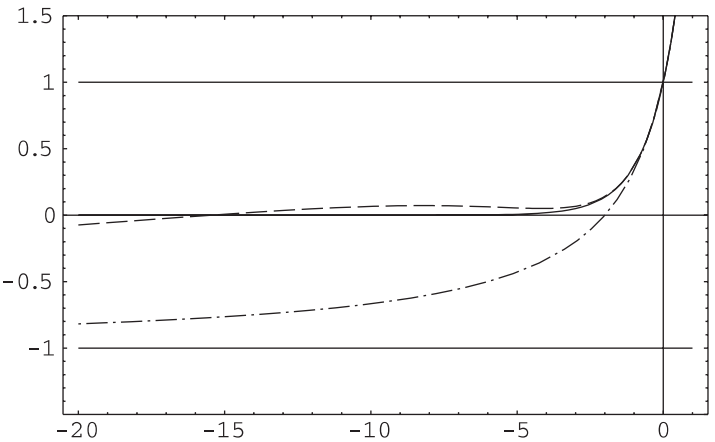

Figure 1. Comparison of linear stability functions ($g(z)$ vs z) for trapezoidal rule (dot–dashed line), RE algorithm with $r = 3$ (dashed line) and exact amplification factor (continuous line).

the real ($z, g(z) \in \mathbb{R}$) linear stability function is plotted for both, Crank–Nicholson and RE schemes. They are compared with the exact amplification factor $g_e(z) = e^z$.

It is worth commenting that using an odd number of Richardson's sub-steps (with $\alpha = \frac{1}{2}$), the stability properties improve for increasing values of r; something similar occurs for r even, but now the absolute stability region fills only an ellipse, which is symmetric and elongated with respect to the real axis and tangent to it for consistency reasons, which growths for r increasing.

A method is said to be *Hopf regular* (Reference [5]), if the frontier of the closed absolute stability region $\mathscr{S}$ is just the imaginary axis. Thus, RE and REL methods are not strictly Hopf regular, but they nearly are (specially for 'quasi-real' spectra; see Figure 1 in Reference [10]). That means, a bifurcation to a periodic solution occurs for the algorithm as long as it occurs for the original system (1) for nearby values of the governing parameters. This property is important when periodic phenomena are modelled.

3.2. Oscillations

For real eigenvalues λ, Figure 1 illustrates complementary aspects of RE and REL schemes. It is seen how such schemes improve the approximation to the exact linear stability function e^z, in contrast with the standard TR, tending to reduce oscillations originated in negative values of the amplification factor. But to do justice to the trapezoidal rule, one must bear in mind that comparisons should be made for equal computational cost. Solving linear problems, for example, RE methods need four resolutions of the linear system per time step but TR needs only one. It is then reasonable to take the ratio of computational costs as 4:1, i.e. for equal computational cost the ratio of the respective time steps should be also 4:1. But even in this case it can be easily checked that RE remains much less oscillatory, unless the oscillations corresponding to the first three intermediate steps of TR were ignored (certainly, unrealistic in practice!). All this is equivalent to compare $g_{\mathrm{RE}}(z)$ vs $g_{\mathrm{TR}}(\frac{z}{4})$ ($g_{\mathrm{TR}} := g_\alpha|_{\alpha=1/2}$) for the first possibility and $g_{\mathrm{RE}}(z)$ vs $[g_{\mathrm{TR}}(\frac{z}{4})]^4$ for the second. Take for instance $z = -20$. From Figure 1 we have $g_{\mathrm{RE}}(-20) \approx -0.0734$ vs $g_{\mathrm{TR}}(-5) \approx -0.429$; and $g_{\mathrm{RE}}(-20) \approx -0.0734$ vs $[g_{\mathrm{TR}}(-5)]^4 \approx 0.0337$.

Figure 2 confirms and enhances these conclusions for the case of complex spectrum. It depicts shaded contour plots for the argument ϕ of the amplification factors (13) and (14), as functions of $z = \lambda \Delta t \in \mathbb{C}$. If the argument is near π, the algorithm strongly oscillates. The black and white (forbidden) zones, respectively, correspond to conventionally selected ranks of arguments ϕ (estimated from limited numerical experiments), say $-\pi \leqslant \phi \leqslant -\pi/4$ radians and $\pi/4 \leqslant \phi \leqslant \pi$. The gray region roughly indicates the 'allowed' zone of low oscillations. To analyse specific cases, the dimensionless spectrum of the gradient matrix $\mathrm{DF}(U^*)$ of system (1), should be superimposed to Figure 2 upper and lower (spectrum footprint [11]). This spectrum contains relevant information from the original problem and its space discretization. Observe that the scales of both figures were selected with a ratio 4:1. As for equal computational cost the (dimensionless) spectrum of TR suffers an homotecy with the inverse ratio, this is an equivalent way to facilitate comparisons. Contrasting directly the sizes of gray zones in Figure 2 (upper and lower) we conclude about the better performance of RE algorithms. But anyway it is appreciated that, near the origin, some relatively small zones exist where RE algorithms can strongly oscillate in contrast to TR. Graphics like those in Figure 2 (lower) have been obtained for $[g_{\mathrm{TR}}(z/4)]^4$ (not shown here). By no means these results improve those for RE.

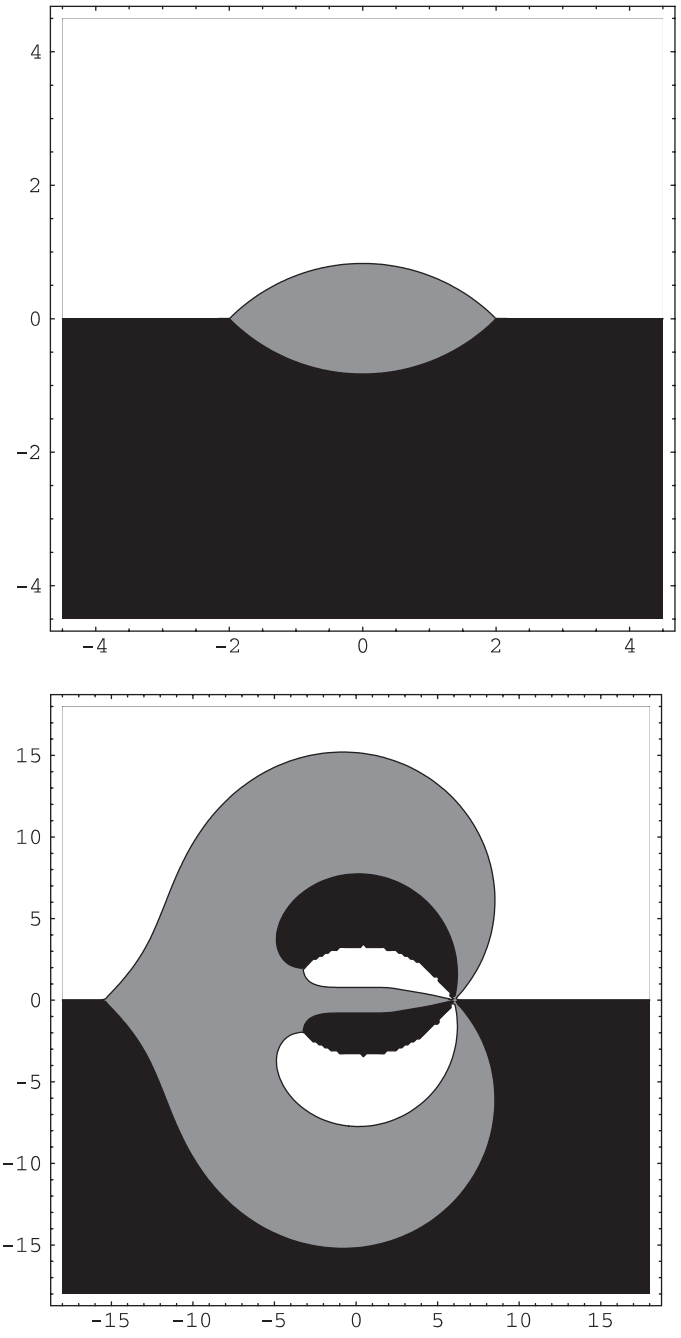

Figure 2. Upper: argument of the amplification factor for the trapezoidal rule, as a function of $z \in \mathbb{C}$. Lower: same for RE and REL algorithms. Black and white: high oscillation zones. Gray: low oscillation zone. Figures are re-scaled to be directly compared for equal computational cost.

3.3. Spurious steady states and period-2 solutions

In this sub-section we briefly discuss two remaining properties of RE and REL time discretizations strategies, pertinent to their genuine non-linear dynamical behaviour: the possibilities of generating spurious fixed points (steady states), and/or period-2 solutions (false transients, jumping between two different states). This issues are relevant for practical calculations. The first possibility can lead to the existence of deceptively smooth spurious steady solutions and also to incorrect stability properties for the true steady states [5, 12]. Normally, there exist many more discrete stable steady states than true stable steady states. Concerning the second possibility, if numerical artifacts are stable, they can drastically affect the performance of the algorithms by reducing their convergence properties owing to the competence between the respective basins of attraction; furthermore, if the period-2 are unstable, it may be that their unstable manifolds could be connected to infinity [12]. That means, there are initial data for which the respective solution blows-up.

As previously commented, for non-linear problems the mapping (3) does not have a unique definition usually. For iterative algorithms, there is *a priori* no guarantee about the root to which iterations would converge. A determination of an effective discrete dynamical system is necessary. Particular determinations can exist leading to algorithms which are not even consistent with the original problem. It is reasonable to ask before what happens with the trapezoidal rule. In Reference [10], a simple scalar example with quadratic second member in (1) is analysed. Thus, two determinations like (3) exist. It is seen that the first one is not consistent with (1), but the second effectively is. The definition domain of both branches as functions of Δt, is limited. The original problem (1) has two equilibrium points, which may not be correctly reproduced by the scheme. But this example shows that for a given exact equilibrium and a given time step Δt, there exists always an algebraic determination such that the equilibrium is exactly inherited. This very simple 1D example illustrates how many difficulties can be introduced by non-linear implicit schemes.

It is trivial to see that every fixed point of the α-method (8) is necessarily an equilibrium of the starting system (1) i.e. no spurious equilibria can be generated (those tending to infinity for increasing time steps; [7, Theorems 6.3.4 and 6.3.6]). Conversely, for any equilibrium point of the original system (1), there corresponds a fixed point of the α-method (8). Let $\mathscr{E} := \{y^* | F(y^*) = 0\}$ be the set of equilibria of (1), and $\mathscr{E}^* := \{U^* | \textit{satisfying } (8) \textit{ with } U_n = U_{n+1} = U^*\}$ the set of fixed points of the α-algorithm (1). These sets are exactly the same: $\mathscr{E} \equiv \mathscr{E}^*$.

Iterations may be prohibitive and unrealistic for the 3D modelling of actual interest. We discuss now a scheme which needs only one iteration per time step: the REL scheme. For the linearized α-method (12), which is almost always correctly defined, it is immediate to show that every fixed point corresponds to an equilibrium of the initial system (11). Conversely, an easy calculation indicates that starting from an exact equilibrium of (1), $y_0 = y^*$, one obtains y_1 such that,

$$[\mathbf{I} - \alpha \Delta t \mathrm{DF}(y^*)](y_1 - y^*) = 0 \tag{15}$$

where $\mathbf{I}$ designates the identity matrix. Thus, $y_1 = y^*$ if $1/\alpha\Delta t$ is not in the spectrum of $\mathrm{DF}(y^*)$. But this condition is nothing more than a particular case of the defining condition for

the recursive rule (12) (the determinant of the matrix of the linear system to be solved in each time step, must not vanish).

Unfortunately, only a part of the good TR's properties are inherited by RE and REL schemes. It is reasonable to expect that the algebraic complexity of these methods, coming from the composed mapping in (7), will originate spurious fixed points. No more attention is given to RE, due to its inherent difficulties related at least with the determination of solutions. With reference to REL algorithm, if $2/\Delta t$ and $2/q\Delta t$ do not belong to the spectrum of $\mathrm{DF}(y^*)$, equilibria of (1) are reproduced as fixed points. All this can be explained with help of expressions (7), (12) (with $\alpha = \frac{1}{2}$), (15), and some simple additional considerations. But contrary to the case of the trapezoidal rule, and again owing to the algebraic complexity of definition (7), REL scheme can generate new fixed points which are false equilibria of (1). In Reference [10], this is illustrated with the previous quadratic example. So we have only $\mathscr{E} \subset \mathscr{E}^*$. According to definitions introduced in Reference [5], REL scheme is *irregular of degree* 1 because not every fixed point corresponds to an exact equilibrium of (1).

To conclude we briefly discuss the possible presence of period-2 solutions. An algorithm is said to be *regular of degree* 2 [5], if it does not admit such transient solutions. It can be easily proved that TR belongs to this category [12]. This is not true for the linearized trapezoidal rule, except for the frequent cases for which $F(U)$ is a quadratic polynomial [10]. But if $F(U)$ is cubic a simple counterexample proves that the statement is false [10]. Unfortunately, REL *scheme is not regular of degree* 2, even for $F(U)$ quadratic [10].

3.4. Conservation properties

It is important to know if Richardson's extrapolation procedures inherit or not the eventual conservation properties of the basic algorithm. An answer is expressed by the following

Lemma
Let the starting algorithm $U_{n+1} = f(U_n)$ satisfy the conservation property,

$$\int_\Omega U_{n+1}\,\mathrm{d}\Omega = \int_\Omega U_n\,\mathrm{d}\Omega + \Delta t \int_\Omega S(x, t_{n+\alpha}, U_{n+\alpha})\,\mathrm{d}\Omega + \Delta t \int_{\partial\Omega} \Phi(x, t_{n+\alpha}, U_{n+\alpha})\,\mathrm{d}\sigma, \quad \alpha \in [0, 1],\ n = 1, 2, \ldots \tag{16}$$

where $\Phi(x, t, U)$ is the flux prescribed on the frontier, $S(x, t, U)$ is a source term, and $t_{n+\alpha} := (1 + \alpha)t_n + \alpha t_{n+1}$, $U_{n+\alpha} := (1 + \alpha)U_n + \alpha U_{n+1}$. Then, for Richardson's extrapolated schemes like (7) it holds,

$$\int_\Omega U_{n+1}^{\mathrm{REX}}\,\mathrm{d}\Omega = \int_\Omega U_n^{\mathrm{REX}}\,\mathrm{d}\Omega + \frac{1}{1 - q^p}\sum_{s=1}^{r} q\Delta t \int_\Omega S(x, t_{n+(s-1+\alpha)q}, U_{n+(s-1+\alpha)q}^{(q\Delta t)})\,\mathrm{d}\Omega - \frac{q^p}{1 - q^p}\Delta t \int_\Omega S(x, t_{n+\alpha}, U_{n+\alpha}^{(\Delta t)})\,\mathrm{d}\Omega$$

$$+\frac{1}{1-q^p}\sum_{s=1}^{r} q\Delta t\int_{\partial\Omega}\Phi(x,t_{n+(s-1+\alpha)q},U^{(q\Delta t)}_{n+(s-1+\alpha)q})\,\mathrm{d}\sigma$$

$$-\frac{q^p}{1-q^p}\Delta t\int_{\partial\Omega}\Phi(x,t_{n+\alpha},U^{(\Delta t)}_{n+\alpha})\,\mathrm{d}\sigma,\quad \alpha\in[0,1],\quad n=1,2,\ldots \tag{17}$$

where REX holds for RE, REL, or other possible extrapolation methods.

Proof
Formula (7) may be rewritten as

$$U^{\mathrm{REX}}_{n+1}=(1-q^p)^{-1}(U^{(q\Delta t)}_{n+rq}-q^pU^{(\Delta t)}_{n+1})\quad (rq=1!) \tag{18}$$

Now we proceed to integrate (18) in the domain Ω. Using (16) to obtain the r-iterated integral with time step $q\Delta t$ we get,

$$\int_\Omega U^{(q\Delta t)}_{n+rq}\,\mathrm{d}\Omega=\int_\Omega U^{\mathrm{REX}}_n\,\mathrm{d}\Omega+q\Delta t\sum_{s=1}^{r}\int_\Omega S(x,t_{n+(s-1+\alpha)q},U^{(q\Delta t)}_{n+(s-1+\alpha)q})\,\mathrm{d}\Omega$$

$$+q\Delta t\sum_{s=1}^{r}\int_{\partial\Omega}\Phi(x,t_{n+(s-1+\alpha)q},U^{(q\Delta t)}_{n+(s-1+\alpha)q})\,\mathrm{d}\sigma \tag{19}$$

Replacing (19) and (16) (resp. for $\int_\Omega U^{(q\Delta t)}_{n+1}\,\mathrm{d}\Omega$ and $\int_\Omega U^{(\Delta t)}_{n+1}\,\mathrm{d}\Omega$) in (18), and reaccommodating, the result (17) is obtained. □

If the source S and the flux Φ are independent of t and U, expression (17) immediately reduces to a form like (16) with S and Φ constants in time. In the general case, the conservation properties of Richardson's extrapolated methods should be understood in the sense of (17), where a more sophisticated and precise evaluation of the contributions of source and surface flux in time is now obtained by weighing with factors $1/(1-q^p)$ and $-q^p/(1-q^p)$, summing 1. This guarantees the order Δt^{p+1}. In other words, if Richardson extrapolated schemes are to be used to approximate originally conservative problems, with starting algorithm (3) also conservative, for the correct physical balance of equations the sources as well as fluxes must be evaluated each one by weighing calculations with time steps $q\Delta t$ and Δt with the aforementioned factors, exactly as indicated in expression (17).

Remark
For original problems written in conservation form the finite element discretization is naturally conservative, unless the unaware user spoils this property by using elements whose shape functions do not sum 1, or unappropriated boundary fluxes in nodes where Dirichlet conditions are prescribed. Common practice is to use these discretized versions of the constitutive law relating fluxes with potentials. If instead such fluxes are obtained via the natural Galerkin formulation at the Dirichlet boundary nodes, conservation properties are restored (see Reference [13]). The conventional practice is just to discard these Dirichlet boundary equations, but such Galerkin equations are precisely the key for the correct balance.

3.5. Linear advection–diffusion tests

If c is the (constant) transport velocity of the linear advection–diffusion operator, v the diffusivity and Δx a typical length for space discretization, the CFL and local Péclét numbers are defined by $\mathscr{C} = c\Delta t/\Delta x$ and $p_e = c\Delta x/v$. In general we can say that after various tests made with trapezoidal rule and REL algorithms at equal computational cost (i.e. $\mathscr{C}_{\mathrm{TR}} = \frac{1}{4}\mathscr{C}_{\mathrm{REL}}$), no essential differences have been found between both performances, excepting some additional dispersive behaviour observed for REL.

First we discuss dispersive properties. Classical 1D tests comparing dispersion relations have shown that TR is somewhat less dispersive than REL, and that its behaviour improves for increasing values of p_e [10]. Due to the practical importance of upwinding for modelling transport phenomena, streamline upwind Petrov Galerkin (SUPG) [14] for the advection term was included in some of the tests. Figures 3 depict a classical numerical experiment of a marching planar, Gaussian-like perturbation, which confirms predictions of the above commented tests. This is a 1D advection–diffusion problem solved in a 2D mesh. The results were obtained for equal computational costs ($\mathscr{C}_{\mathrm{TR}} = \frac{3}{4}$, $\mathscr{C}_{\mathrm{REL}} = 3$, $p_e = 2.67$) [10]. Dispersive effects for REL method can be appreciated at $t = 10$ (Figure 3 lower), originated by short waves soon extinguished. Figure 4 shows results for a similar experiment, but now three dimensional. Even for $t \approx 50$ strong dispersive rear oscillations are observed, specially for REL algorithm, according to predictions. It is seen that three dimensionality involves intense dispersion effects. Here, quasi-uniform two- and three-dimensional meshes of finite elements with linear form functions were used.

Concerning diffusive properties, 1D tests show very similar performances for TR and REL schemes [10]. This situation basically persists for different values of the involved parameters. Marching perturbation experiments also confirm this conclusion (Figures 3 and 4).

The effect of mesh refinement on numerical diffusivity was also tested [10]. This is an important point, because the numerical solution of any minimally interesting real problem to be modelled is inconceivable without using non-trivial adaptive meshes. To this purpose, a one-dimensional example solved in a bounded domain of length 6 was considered. Results for two meshes were compared: a uniform mesh of 200 1D linear finite elements ($\Delta x = 0.03$), and a second mesh with the same number of elements, but with a highly refined thin zone, located between $x = 2.4$ and 2.7, and separating two zones of similar coarse elements. The degree of distortion, i.e. the ratio of maximum to minimum element sizes, was 100. For the regular mesh, the values of the main parameters were: $\mathscr{C} = 2.666\ldots$, $p_e = 0.3$. Unfortunately, higher values of p_e (and/or $\mathscr{C}$) originate severe difficulties to obtain eigenvalues numerically [15]. Exact formulae to calculate eigenvalues of (irregular) matrices associated to non-uniform meshes, are certainly not available. Thus, we have to restrict our analysis to less interesting cases. Such results predicted, (1) that refinement deteriorates the performances of both schemes (this is to be expected, owing to the increasing condition number of the involved matrices originated by mesh refinement); (2) that REL presents a slightly better behaviour compared with TR. Numerical experiments (results not shown here), involving a marching planar Gaussian, have shown no essential differences between performances of TR and REL methods with distorted meshes. The Gaussian, initially located in the left coarse zone ($x = 0.5$), moves to the right crossing the densified layer towards the other coarse zone. This is a plane problem solved in a quasi-uniform and in a refined 2D meshes, both with linear triangles, the first containing 18 000 elements (mean size ≈ 0.03) and the second, 56 000 elements (maximal size ≈ 0.3, minimal size ≈ 0.003).

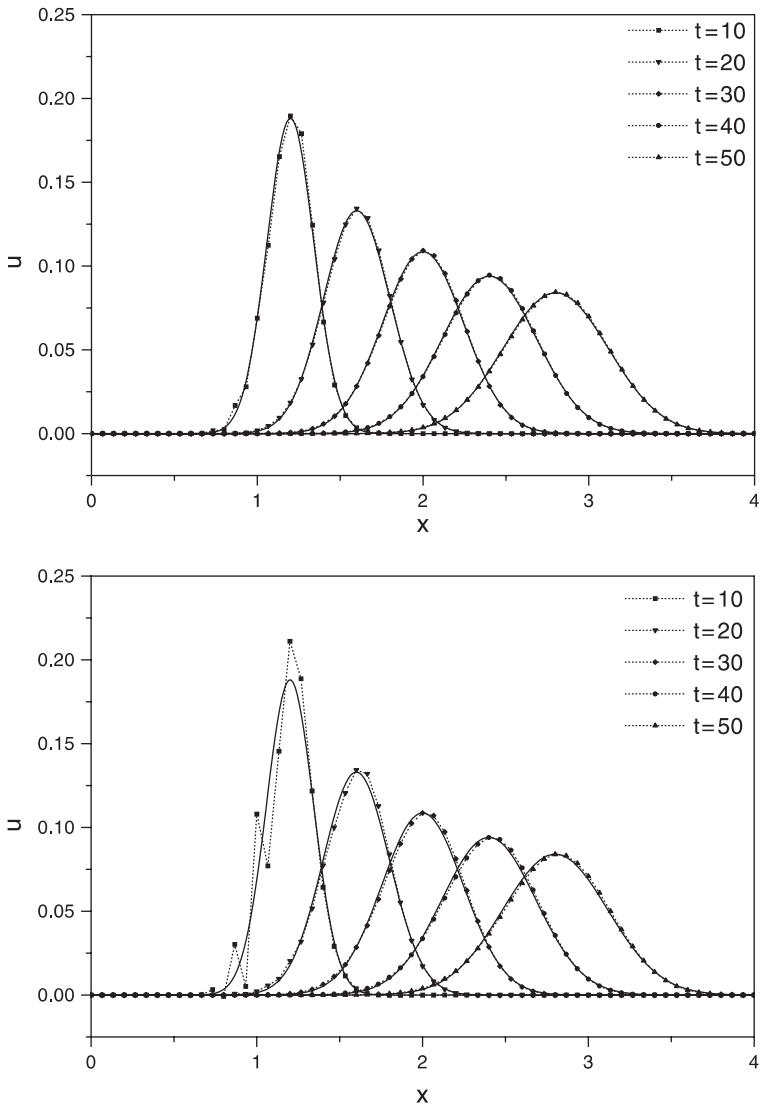

Figure 3. Two-dimensional experiment: decay of an initial planar and local perturbation. Values of the solution for $t = 10, 20, 30, 40, 50$. Upper: trapezoidal rule, at equal computational cost ($\mathscr{C} = \frac{3}{4}$, $p_e = 2.67$). Lower: REL algorithm ($\mathscr{C} = 3$, $p_e = 2.67$). Rear dispersive effects are observed for REL. Exact solution is indicated by a continuous line. SUPG is included.

4. STRATEGIES FOR TIME STEP ADAPTATION

The presentation of the REL algorithm made in the previous sections, leaves complete freedom for the choice of any kind of strategy to control the time step. As for the computational

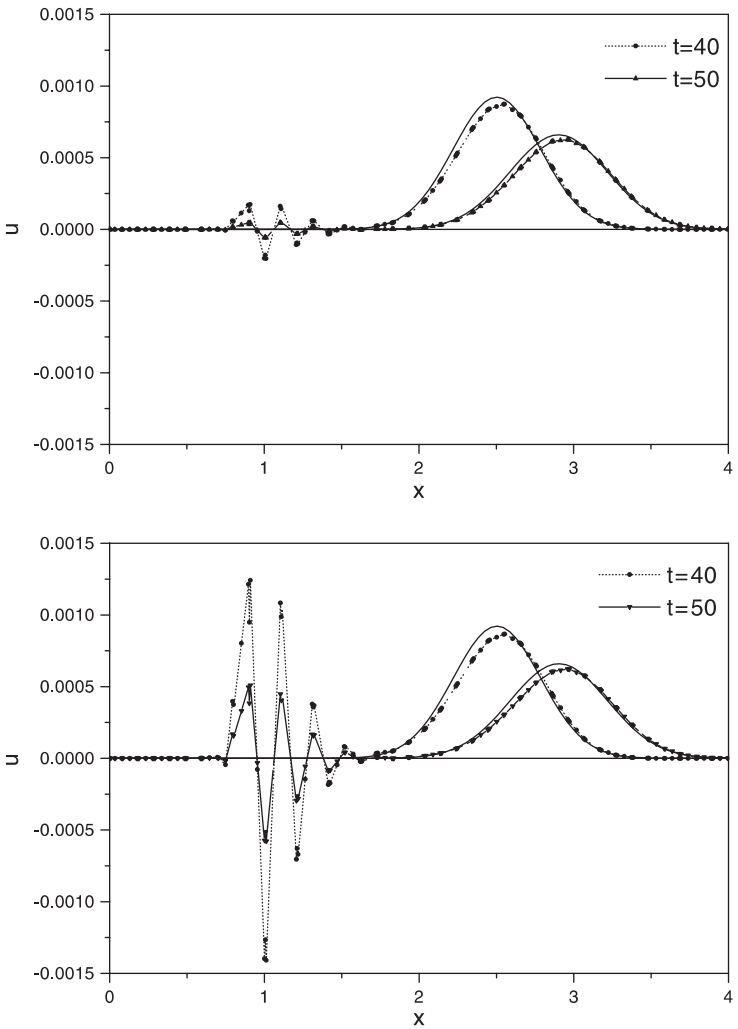

Figure 4. Three-dimensional experiment: decay of an initial local perturbation. Values of the solution (at the plane of symmetry) for $t = 40$ and 50. Upper: Trapezoidal rule, at equal computational cost ($\mathscr{C} = \frac{3}{4}$, $p_e = 2.67$). Lower: REL algorithm ($\mathscr{C} = 3$, $p_e = 2.67$). Rear dispersive effects are observed in both cases. Exact solution is indicated by a continuous line. SUPG is included.

implementation of REL one needs simultaneously to programme the Newton–Raphson method, the possibility of making more than one iteration remains also opened. Thus, a lot of (perhaps well known) strategies are at hand to control errors. For only one iteration, the time step can be adapted. If this is not enough to guarantee the desired level of errors originated by severe non-linearities, one (or more) additional iterations can be made, using expression (9) instead of (12). Also combinations of these two possibilities may be imagined, as well as the use of the

fully converged REL method ($v \to \infty$). It is worth commenting that in some of the applications to groundwater flow modelling, cases have been observed for which with only one iteration the error level was smaller than for converged iterations.

For time-step adaptation a viable strategy is now described. As it was mentioned in the previous section, the proposed extrapolated algorithm is order $\mathcal{O}(\Delta t^3)$. The method provides information about the local truncation error. This information can be employed as an error indicator, to control the calculation precision, keeping the time integration error within an acceptable pre-determined band. This can be done by appropriately modifying the time step size, decreasing/increasing it when the error goes beyond/below the upper/lower bound, respectively. The standard strategy to adapt the time step is to reduce it to a half when the error goes beyond the upper bound or to double it when it goes below the lower bound. As the method is $\mathcal{O}(\Delta t^3)$, reducing the time-step size to the half will reduce the local error by a factor eight. When the time step is modified, several computations must be redone with the consequent computational costs; therefore the band precision width should be large enough (for a certain factor larger than the mentioned factor eight) in order to avoid increase/decrease cycles of the time step. In our case we use a value $r = 3$ for the Richardson extrapolation. For that reason it is convenient to decrease the time step in a factor three instead of two. In this way, it is possible to use again at least some of the computations already done. Then, when the time step is reduced, the local error will decrease in a factor 27 ($\frac{1}{33}$). Therefore, the error band should defined as follows: upper bound = user tolerance; lower bound = user tolerance/54.

Another possible strategy is that described for a particular example at the end of the next section. As it will be seen later, the original algorithm is $\mathcal{O}(\Delta t)$ for this example. Thus, Richardson's extrapolation rises this order to $\mathcal{O}(\Delta t^2)$.

5. AN APPLICATION TO VARIABLY SATURATED GROUNDWATER FLOW MODELLING

In this section we will examine the application of the proposed algorithm to the highly non-linear Richards equation [16] which is commonly used to describe water flow in unsaturated porous media. This equation can be expressed in terms of both the pressure head h and the volumetric water content θ, resulting in an 'h-based', a 'θ-based' or a 'mixed' form of Richards equation [17]. It is generally accepted that numerical solutions of the mixed form circumvent major mass balance and accuracy problems associated to the high non-linearity of Richards equation and for this reason will be considered in the present example. In a previous work by Guarracino and Quintana [18] a third-order Crank–Nicholson scheme based on Richardson extrapolation was employed to solve the h-based form of Richards equation with promising results.

The mixed form of Richards' equation can be written as [17]

$$\frac{\partial \theta(h)}{\partial t} + \nabla \cdot [\mathbf{K}_s K_r(h) \nabla (h + z)] = 0 \tag{20}$$

where $\mathbf{K}_s$ is the saturated hydraulic conductivity tensor, $K_r(h)$ the relative hydraulic conductivity function and z the vertical co-ordinate (positive upward).

To solve Equation (20) we consider a two-dimensional domain Ω with boundary $\partial\Omega = \Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}}$ ($\Gamma^{\mathrm{D}} \cap \Gamma^{\mathrm{N}} = 0$) and we state a differential problem as follows:

$$\begin{cases} \dfrac{\partial\theta(h)}{\partial t} + \nabla \cdot [\mathbf{K}_{\mathrm{s}} K_{\mathrm{r}}(h) \nabla(h+z)] = 0, & x \in \Omega \\ h(x,t) = h^*(x,t), & x \in \Gamma^{\mathrm{D}} \\ -\mathbf{K}_{\mathrm{s}} K_{\mathrm{r}}(h) \nabla(h+z) \cdot v = Q^*(x,t), & x \in \Gamma^{\mathrm{N}} \\ h(x,0) = h_0(x), & x \in \Omega \end{cases} \tag{21}$$

where h^* denotes the prescribed values of the pressure head over Γ^{D}, v is the unit outer normal to Γ^{N}, Q^* denotes the specified values of normal component of flow on Γ^{N} and h_0 is the initial guess for h. Note that in order to solve problem (21) constitutive models of $\theta(h)$ and $K_{\mathrm{r}}(h)$ are also required.

To develop a finite element procedure for spatial discretization of problem (21), we state the following weak form of the original differential problem:

$$\left(\frac{\partial\theta(h)}{\partial t}, v\right) + a(h,v) = l(v) \tag{22}$$

where

$$\begin{aligned} \left(\frac{\partial\theta(h)}{\partial t}, v\right) &= \int_\Omega \frac{\partial\theta(h)}{\partial t} v \, \mathrm{d}x \\ a(h,v) &= \int_\Omega \mathbf{K}_{\mathrm{s}} K_{\mathrm{r}}(h) \nabla h \cdot \nabla v \, \mathrm{d}x \\ l(v) &= -\int_{\Gamma^{\mathrm{N}}} Q^* v \, \mathrm{d}x - \int_\Omega \mathbf{K}_{\mathrm{s}} K_{\mathrm{r}}(h) \nabla z \cdot \nabla v \, \mathrm{d}x \end{aligned} \tag{23}$$

with $v \in V$, $V = \{v \in H^1(\Omega)$ such that $v = 0$ on $\Gamma^{\mathrm{D}}\}$.

Without going through the details, the Galerkin finite element method applied to (22) yields the system of non-linear differential equations

$$\mathbf{M}\frac{\partial\boldsymbol{\theta}(\mathbf{h})}{\partial t} + \mathbf{A}(\mathbf{h})\mathbf{h} = \mathbf{B}(\mathbf{h}) \tag{24}$$

where $\mathbf{h}$ and $\boldsymbol{\theta}$ are, respectively, the vectors of unknowns (pressure head) and water content values at each node of the finite element mesh. The components of matrices $\mathbf{M}$ and $\mathbf{A}$, and vector $\mathbf{B}$ are of the form

$$M_{ji} = (\phi_i, \phi_j), \quad A_{ji} = a(\phi_i, \phi_j), \quad B_j = l(\phi_j) \tag{25}$$

where ϕ_j is the linear Lagrange basis function corresponding to node j. The non-linear integrals in (25) are evaluated using a second-order Gaussian quadrature.

For time discretization we use a two-time level finite difference scheme. An internal α-method of system (24) can be expressed as

$$\mathbf{M}\frac{\boldsymbol{\theta}_{n+1}-\boldsymbol{\theta}_n}{\Delta t}+\mathbf{A}_{n+\alpha}\mathbf{h}_{n+\alpha}=\mathbf{B}_{n+\alpha} \tag{26}$$

where $\mathbf{h}_{n+\alpha}=(1-\alpha)\mathbf{h}_n+\alpha\mathbf{h}_{n+1}$, $\boldsymbol{\theta}_{n+1}=\boldsymbol{\theta}(\mathbf{h}_{n+1})$, $\boldsymbol{\theta}_n=\boldsymbol{\theta}(\mathbf{h}_n)$, $\mathbf{A}_{n+\alpha}=\mathbf{A}(\mathbf{h}_{n+\alpha})$ and $\mathbf{B}_{n+\alpha}=\mathbf{B}(\mathbf{h}_{n+\alpha})$.

Finally, system (26) is linearized using a Newton–Raphson method as explained in previous sections. The resulting algorithm can be written in the form

$$\mathbf{h}_{n+1}^{(v+1)}=\mathbf{h}_{n+1}^{(v)}-[D\boldsymbol{\varphi}(\mathbf{h}_{n+1}^{(v)})]^{-1}\boldsymbol{\varphi}(\mathbf{h}_{n+1}^{(v)}) \tag{27}$$

where the residual and tangent matrix have, respectively, the following expressions:

$$\begin{aligned}\boldsymbol{\varphi}(\mathbf{h}_{n+1}^{(v)})&=\mathbf{M}\frac{\boldsymbol{\theta}_{n+1}^{(v)}-\boldsymbol{\theta}_n}{\Delta t^{n+1}}+\mathbf{A}_{n+\alpha}^{(v)}\mathbf{h}_{n+\alpha}^{(v)}-\mathbf{B}_{n+\alpha}^{(v)}\\ D\boldsymbol{\varphi}(\mathbf{h}_{n+1}^{(v)})&=\mathbf{M}\frac{\dot{\boldsymbol{\theta}}_{n+1}^{(v)}}{\Delta t^{n+1}}+\alpha(D\mathbf{A}_{n+\alpha}^{(v)}\mathbf{h}_{n+\alpha}^{(v)}+\mathbf{A}_{n+\alpha}^{(v)}-D\mathbf{B}_{n+\alpha}^{(v)})\end{aligned} \tag{28}$$

To obtain the linearized α-algorithm we perform only one iteration of (27) using as initial guess the solution of the previous time step (that is $\mathbf{h}_{n+1}^{(0)}=\mathbf{h}_n$). It is important to remark that for the particular case of Richards equation the linearized Crank–Nicholson algorithm ($\alpha=\frac{1}{2}$) is $\mathcal{O}(\Delta t)$ [19]. This first-order accuracy of the linearized scheme motivates the use of Richardson extrapolation to recover the second-order accuracy. This non-iterative second-order strategy is an attractive alternative to the conventional Newton and Picard schemes commonly used for solving Richards equation.

The proposed linearized algorithm was applied to three test problems involving unsaturated and variably saturated flows. Numerical tests were designed to verify the second-order accuracy, the mass conservation and the computational efficiency of the algorithm. In test 3 we also include a dynamic time step control strategy which significantly improve the numerical performance.

5.1. Test 1

The objective of the present test is to verify the level of accuracy of the proposed time discretization strategy by comparing numerical and analytical solutions. For the numerical analysis we employ a one-dimensional analytical solution of Richards equation obtained for restricted initial and boundary conditions.

The constitutive relations considered in this test are $\theta(h)=\mathrm{e}^{\alpha h}$ and $K_\mathrm{r}=\mathrm{e}^{(\gamma+1)\alpha h}$, where γ and α are models parameters. Next, we state the following differential problem which represents

an infiltration front moving into a completely dry soil:

$$\begin{cases} \dfrac{\partial\theta(h)}{\partial t} + \dfrac{\partial}{\partial z}\left[K_s K_r(h)\dfrac{\partial}{\partial z}(h+z)\right] = 0, \quad z \in [0, L] \\ h(0,t) = -\infty \\ h(L,t) = \dfrac{1}{\gamma\alpha}\ln\left[\dfrac{A}{K_s}(1 - e^{-\gamma\alpha A t})\right] - L \\ h(z,0) = -\infty \end{cases} \tag{29}$$

here L is the length of the spatial domain and A a parameter related to the amplitude of the saturation front. The analytical solution of problem (29) is [18, 20]

$$h(z,t) = \begin{cases} \dfrac{1}{\gamma\alpha}\ln\left[\dfrac{A}{K_s}(1 - e^{-\gamma\alpha(At - L + z)})\right] + L, & z > L - At \\ -\infty, & z \leqslant L - At \end{cases} \tag{30}$$

valid for $t \in (0, L/A]$.

Although analytical solution (30) is one dimensional, we compute numerical solutions on a two-dimensional domain by considering no flow conditions in lateral boundaries. Thus we consider a rectangular domain of 1 by 12 units of length in directions x and z, respectively. Non-dimensional values of the hydraulic parameter were taken to be $K_s = 1$, $\alpha = 0.01$, $\gamma = 3$ and $A = 2$. Numerical solutions were obtained on a coarse unstructured triangular mesh of 245 nodes.

The completely dry initial condition of the test problem ($h(z,0) = -\infty$) is a drawback in the comparison of exact and numerical solutions. To obtain a numerical solution, a finite value of the pressure head has to be considered. This approximated initial condition implies a residual water content in the porous medium introducing an additional source of numerical error which has to be taken into account in the analysis of the numerical results.

In Figure 5 the comparison of analytical and numerical solutions at times 2.5 and 5 are shown. Numerical values correspond to a vertical profile situated in the mid point of the simulation domain.

To evaluate the accuracy of the time scheme, an error norm is computed as follows:

$$e(\Delta t) = \frac{1}{N_p}\sum_{k=1}^{N_p}(h_k^{N,\Delta t} - h_k^A)^2 \tag{31}$$

where $h_k^{N,\Delta t}$ is the numerical solution at node k obtained with a constant time step Δt, h_k^A denotes the analytical solution, and N_p is the total number of nodal points. To minimize the effect of the approximated initial condition in the computation of (31) we consider only nodes with analytical values $h_k^A > h^{\min} > -\infty$, where $h^{\min}$ is a lower cutoff value for the pressure head.

The error norms of linearized Crank–Nicholson (TRL) and linearized Crank–Nicholson with three-step Richardson extrapolation (REL) schemes are shown in Figure 6. The slopes of the curves show respectively linear and quadratic behaviours verifying the theoretically expected convergence rates.

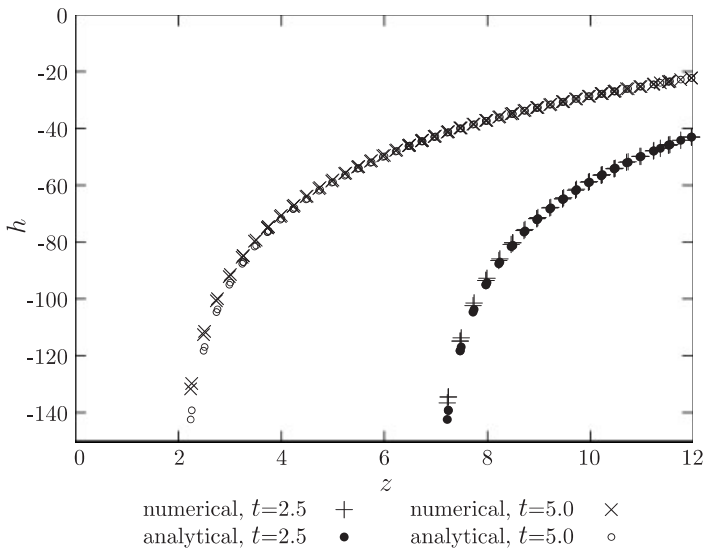

Figure 5. Comparison of analytical and numerical solutions for test case 1.

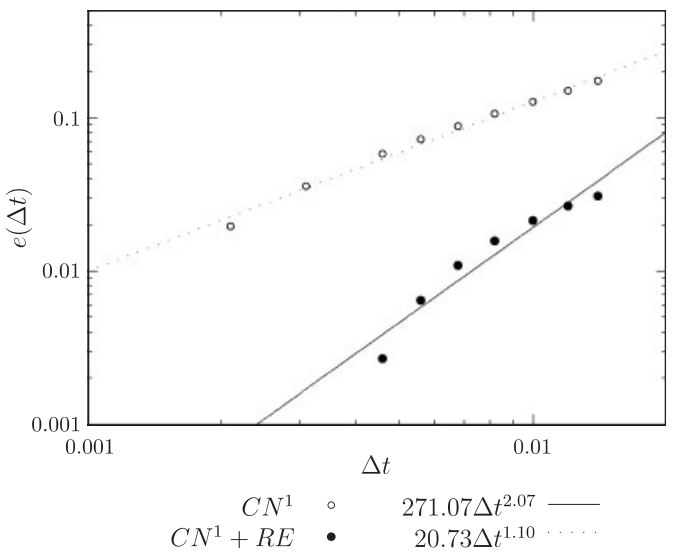

Figure 6. Accuracy of time discretization schemes.

5.2. Test case 2

Large mass balance errors in standard approximations of Richards' equation have been reported by several authors [21, 17]. The reason for large mass balance errors resides in the numerical approximation of the time derivative term. In the mixed form of Richards equation the time

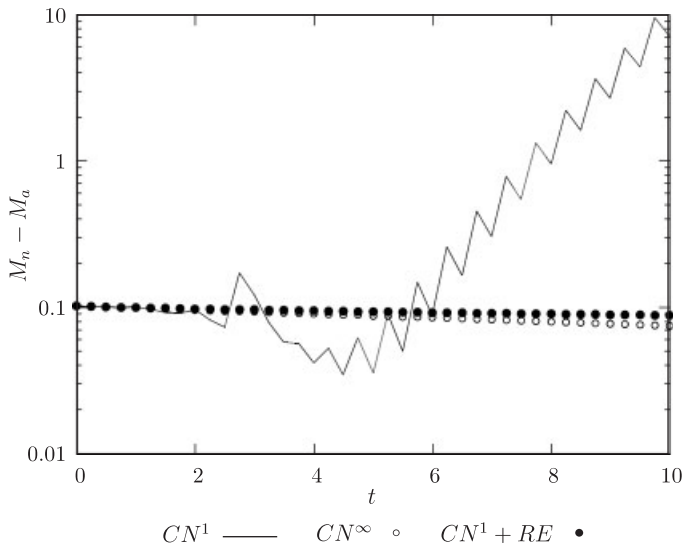

Figure 7. Mass balance error for test case 2.

derivative term is $\partial\theta/\partial t$ while in the h-based form this term is expressed as $C(h)\partial h/\partial t$ with $C(h) = \partial\theta/\partial h$. Both $\partial\theta/\partial t$ and $C(h)\partial h/\partial t$ are mathematically equivalent in the continuous partial differential equation but their discrete analogous are not equivalent. This inequality in the discrete forms is exacerbated by the highly non-linear nature of the constitutive relations, which leads to significant mass balance errors in the h-based schemes. For example, numerical solutions of Richards equation using a backward Euler time discretization produce unacceptably large mass balance errors. This is true for any iteration method (Picard, Newton–Raphson, etc.) and for both finite difference and finite element approximations in space [17].

In the present example we test the ability of the proposed algorithm to conserve numerically the global mass. Assuming a unity value for the coefficient γ of the previous test case, it is possible to obtain the following analytical closed-form expression for time evolution of global mass [18]:

$$M_{\mathrm{a}}(t) = \frac{A}{K_{\mathrm{s}}}\left[At + \frac{1}{\alpha}(\mathrm{e}^{-\alpha At} - 1)\right] \tag{32}$$

To test the algorithm we consider an infiltration case similar to the previous one, but in a rectangular domain of 1 by 25 units of length. The physical and discretization parameters of the problem were taken to be $K_{\mathrm{s}} = 1$, $\alpha = 0.01$, $A = 1$, $N_{\mathrm{p}} = 877$ and $\Delta t = 0.25$. Figure 7 shows the difference between numerical (M_{n}) and analytical (M_{a}) masses as a function of time t for the following time schemes: linearized Crank–Nicholson (TRL), classical Crank–Nicholson (TR^{∞}) and linearized Crank–Nicholson with three-step Richardson extrapolation (REL). The time accuracy of scheme TRL is order one while both TR^{∞} and REL are order two. It can be observed that at $t = 0$ the three schemes have the same error. As mentioned in test case 1, the initial mass in the domain would be zero, but as it is not possible to deal with an exact initial condition ($h = -\infty$), we have a non-zero value for the initial mass. The improvement

of mass conservation with the degree of accuracy in time schemes is clearly observed in the figure. As it was pointed out, the time derivative is crucial in the numerical performance. The TRL scheme is $\mathcal{O}(\Delta t)$ accurate and produce unacceptable large mass balance errors. Similar behaviours were observed for numerical solutions of the h-based form of Richards equation using one order time accurate schemes [17, 18]. Although both TR^{∞} and REL schemes are $\mathcal{O}(\Delta t^2)$ accurate and show good mass balances during the simulation period, the REL scheme is lightly more conservative.

5.3. Test case 3

The objective of this final test is to show the performance of the algorithm to simulate water flow under variably saturated conditions, as well as to introduce a dynamic time step control for improving computational cost.

The numerical test considers the elevation of the water table due to a constant infiltration from the surface in a rectangular domain of 400 cm wide by 300 cm deep. The initial condition is the hydrostatic state with a horizontal water table situated at 200 cm from the soil surface. As boundary condition, a constant infiltration rate of 10^{-5} cm/s is applied to an interval of 200 cm located in the left half of the upper boundary (soil surface).

For the constitutive relations we adopt the well-known van Genuchten model, written as [22]

$$\theta(h) = \begin{cases} \dfrac{\theta_s - \theta_r}{[1 + (\hat{\alpha}|h|)^{\hat{n}}]^{\hat{m}}}, & h < 0 \\ \theta_r, & h \geqslant 0 \end{cases} \tag{33}$$

$$K_r(h) = \begin{cases} \dfrac{[1 - (\hat{\alpha}|h|)^{\hat{n}-1}[1 + (\hat{\alpha}|h|)^{\hat{n}}]^{-\hat{m}}]^2}{[1 + (\hat{\alpha}|h|)^{\hat{n}}]^{\hat{m}/2}}, & h < 0 \\ 1, & h \geqslant 0 \end{cases} \tag{34}$$

where θ_r and θ_s are the residual and saturated water content, respectively; $\hat{\alpha}$ and $\hat{n}$ are model parameters, and $\hat{m} = 1 - 1/\hat{n}$. In this test the following parameter values for a silt soil [23] are considered: $\theta_r = 0.034$, $\theta_s = 0.46$, $\hat{\alpha} = 0.016\,\text{cm}^{-1}$, $\hat{n} = 1.37$, and $K_s = 6.0\,\text{cm/day}$.

Numerical solutions were obtained using a 2D triangular mesh of 1436 nodes and a time-step (Δt) of 15 min. Pressure head contours after 10 and 20 days of simulation are plotted in Figures 8 and 9, respectively. Negative values of h indicate unsaturated regions and positive ones fully saturated zones, while the zero contour line ($h = 0$) corresponds to the position of the water table. Figure 8 shows an infiltration front moving towards the water table in the left side of the computational domain. After 20 days of simulation the infiltration front has reached the saturated zone and a water table elevation of approximately 80 cm is observed on the left side of the domain (Figure 9). This test shows that the proposed algorithm is appropriate to deal with problems where the water table is not fixed during the simulation period. From a numerical point of view this implies that the domain region where the non-linearities of Richards' equation take place (unsaturated region) is changing with time.

To improved the computational efficiency of the algorithm we implement a dynamic time step control where the time step size is increased or decreased depending on the value of an internal relative error. This control is based on the fact that at time t^{n+1} the algorithm

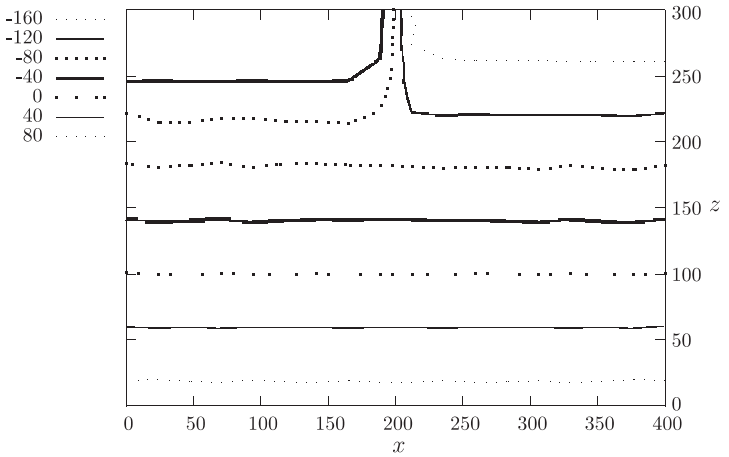

Figure 8. Pressure head contours at $t = 10$ days for test case 3.

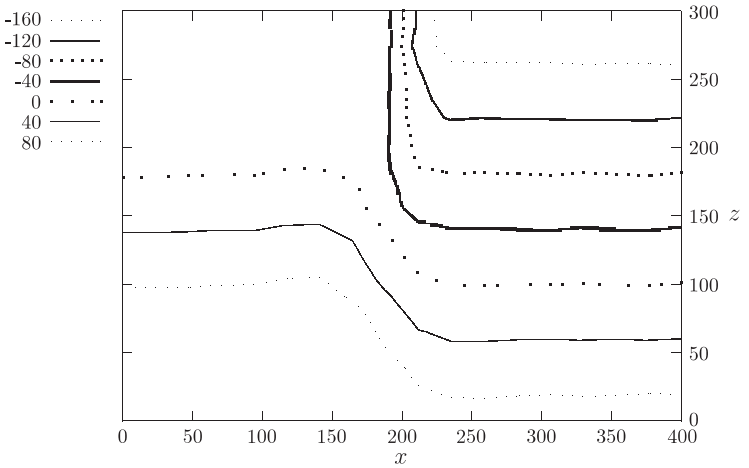

Figure 9. Pressure head contours at $t = 20$ days for test case 3.

computes both the numerical solution $h_{n+1}^{\Delta t}$ using a single time step Δt and the extrapolated solution $h_{n+1}^{\mathrm{RE},\,\Delta t/3}$, which is more accurate than the former. Then we can define an internal relative error between both solutions:

$$e_{\mathrm{r}}(\Delta t) = |h_{n+1}^{\Delta t} - h_{n+1}^{\mathrm{RE},\,\Delta t/3}|_{L^\infty} \tag{35}$$

By imposing a tolerance ε_{a} on $e_{\mathrm{r}}(\Delta t)$ it is possible to state a dynamic time step control strategy as follows: Δt is increased by a prescribed factor if $e_{\mathrm{r}}(\Delta t) < \varepsilon_{\mathrm{a}}$ and it is decreased whenever this condition is not satisfied. This automatic time step adjustment is easy to implement and minimal additional calculus are required. In the present test the time step is increased by a factor 2 when $e_{\mathrm{r}}(\Delta t) < \varepsilon_{\mathrm{a}}$ and it is decreased by a factor $\frac{1}{3}$ when $e_{\mathrm{r}}(\Delta t) > \varepsilon_{\mathrm{a}}$.

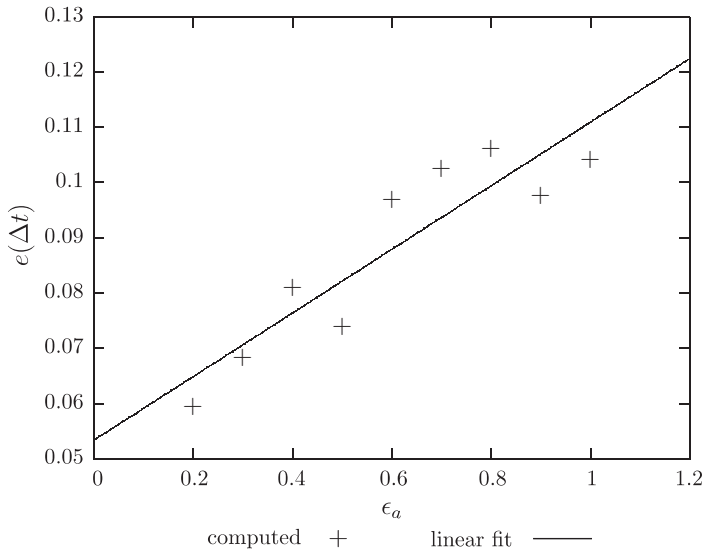

Figure 10. Accuracy of the time step adjustment scheme.

In order to analyse the accuracy of the time step control strategy we consider as 'exact' solution of this test a third-order accurate numerical solution obtained using the extrapolated algorithm with a classical Newton–Raphson linearization. The error in the numerical approximation is computed by using this 'exact' solution in formula (31). As Figure 10 shows, the numerical error increase linearly as the tolerance ε_a get greater.

The computational efficiency is measured on the basis of the number of linearized problems solved during the simulation period, for a given tolerance. Figure 11 shows the computational efficiency for different values of tolerance ε_a which is approximately proportional to $1/\sqrt{\varepsilon_a}$. Comparing Figures 10 and 11 we can conclude that the computational cost can be drastically reduced without loosing much precision in the numerical approximation. The number of problems solved without time step adaptation is 1920 and can be reduce to 779 using the proposed adaptation strategy with a tolerance $\varepsilon_a = 0.5\,\text{cm}$. Moreover, the number of problems solved using a classical Crank–Nicholson scheme is 7691, which is approximately one order larger than the obtained with the proposed algorithm and time step adaptation. Thus, for this case *one order of magnitude for efficiency was gained*. Note that *both* schemes have the same accuracy ($\mathcal{O}(\Delta t^2)$), because now the linearized Crank–Nicholson algorithm is only $\mathcal{O}(\Delta t)$.

6. CONCLUSIONS

A promising non-iterative algorithm (REL) based on the linearized trapezoidal rule together with Richardson's extrapolation, has been studied and tested. An illustrative application to a 2D highly non-linear problem (variably saturated groundwater flow modelling) has been also presented. The proposed scheme, having a wide scope for 3D applications, has shown to be a good candidate to efficiently solve problems with large variations in time, reducing

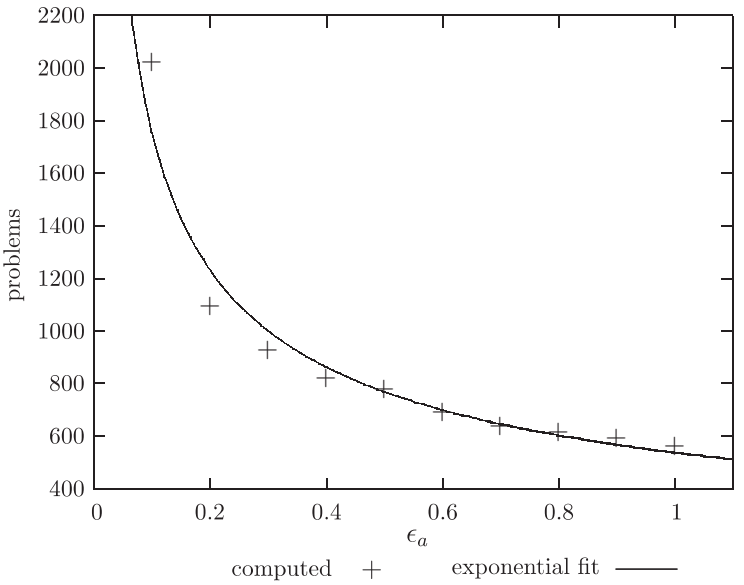

Figure 11. Computational efficiency of the time step adjustment scheme.

significantly the computational cost of numerical simulations. Other specific conclusions of this analysis are:

- Even when the linearized Crank–Nicholson scheme is unconditionally stable, two-steps REL algorithm is not. But this nice property is practically recovered if one uses three intermediate steps for Richardson's extrapolation; and completely recovered, if additionally the spectrum of the gradient matrix DF is real.
- Bifurcations to periodic solutions occur for REL scheme as long as they occur for the original system (1), for nearby values of the governing parameters. That means, the scheme is not strictly *Hopf regular* [5], but it nearly is.
- For equal computational costs REL scheme tends to correct the bad tendency to oscillate, characteristic of the trapezoidal rule.
- Unfortunately, owing to the notorious algebraic complexity introduced by Richardson's extrapolation procedure, excellent properties of the trapezoidal rule are lost: undesirable spurious steady states can be created by REL scheme, as well as non-physical transients like period-2 solutions. These issues are relevant for practical calculations. The first possibility can lead to the existence of deceptively smooth spurious steady solutions looking as real, and also to incorrect stability properties for the true steady states. Concerning the second possibility, periods-2 can drastically affect the performance of the algorithms by reducing their convergence properties, owing to the competence originated between false and real basins of attraction.
- The Richardson's extrapolated algorithm is conservative as long as it is the basic scheme. This feature of the algorithm allows to deal with problems whose conservation properties are important to preserve accurately like the example shown in Section 5.

- No evident differences have been observed in relation to numerical diffusivity properties for TR and REL algorithms.
- Severe mesh refinement gradients deteriorate diffusive characteristics of TR and REL schemes, essentially in a similar way.
- An application has been presented showing that the proposed (non-iterative) REL algorithm, together with a dynamic time-step control strategy, can improve efficiency approximately by an order of magnitude with respect to the classical (iterative) Crank–Nicholson method.

Future work will be oriented to investigate the performance of REL in the field of computational fluid dynamics. In particular, 3D modelling using the Navier–Stokes equations as well as applications to non-Newtonian hæmodynamics.

ACKNOWLEDGEMENTS

The authors are indebted to Marcelo Alvarez for calculations made for Figures 3 and 4.

This work was partially supported by ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica) under PICT 12-03239.

REFERENCES

1. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**:337–357.
2. George P, Hecht F, Saltel E. Fully automatic mesh generator for 3D domain of any shape. *IMPACT of Computer Science and Engineering* 1990; **2**:187–218.
3. Módena R, Vénere MJ, Quintana F. An analysis of schemes for temporal integration in transient diffusion–reaction problems (in Spanish). In *Mecánica Computacional*, Dari E, Padra C, Saliba R (eds), vol. XVIII. 1991; 867–876.
4. Quintana F, Guarracino L, Bevilacqua AM. Analysis of transport and diffusion of radionucleids in deep rock formations (in Spanish). In *Mecánica Computacional*, Idelsohn S, Sonzogni V, Cardona A (eds), vol. XXI. 2002; 2385–2394.
5. Iserles A, Peplow AT, Stuart AM. An unified approach to spurious solutions introduced by time discretisation. Part I: basic theory. *SIAM Journal on Numerical Analysis* 1991; **28**:1723–1751.
6. Stuart AM, Peplow AT. The global dynamics of discrete semilinear parabolic equations. *SIAM Journal on Numerical Analysis* 1993; **30**:1622–1663.
7. Stuart AM, Humphries AR. In *Dynamical Systems and Numerical Analysis*, Ciarlet PG, Iserles A, Kohn RV, Wright MH (eds), Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press: Cambridge, 1996.
8. Henrici P. *Discrete Variable Methods in Ordinary Differential Equations*. Wiley: New York, London, Sydney, 1962.
9. Basombrío FG. On linearized stability and nonlinear realities. *Internal Report CAB-CNEA*, 2000 (See http://cabmec1.cnea.gov.ar/ ∼fgb/riratrl/riratrl.ps.gz)
10. Basombrío FG. A third-order Crank–Nicholson algorithm based on Richardson's extrapolation. General properties. *Internal Report Centro Atómico Bariloche* (*CAB*)-*CNEA*, 2004 (See http://cabmec1.cnea.gov.ar/ ∼fgb/extrap/extrap_inf.pdf)
11. Morton KW. *Numerical Solution of Convection–Diffusion Problems*. Chapman & Hall: London, 1980.
12. Elliot CM, Stuart AM. The dynamics of the θ-method. *SIAM Journal on Scientific and Statistical Computing* 1991; **12**:1351–1372.
13. Lynch DR. Mass conservation in finite element groundwater models. *Advances in Water Resources* 1984; **7**:67–75.
14. Hughes TJR. *The Finite Element Method*. Prentice-Hall: Englewood Cliffs, NJ, 1987.

15. Basombrío FG. An example showing severe difficulties in the evaluation of eigenvalues of simple tridiagonal non-symmetric matrices. *Proceedings MECOM'99* (CD-Rom), Computational Mechanics Congress: Mendoza, Argentina, 6–10 September 1999 (See http://cabmec1.cnea.gov.ar/∼fgb/autopar/autopar.ps)
16. Richards L. Capillary conduction of liquids through porous mediums. *Physics* 1931; **1**:318–333.
17. Celia MA, Bouloutas ET, Zarba RL. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research* 1990; **26**:1483–1496.
18. Guarracino L, Quintana JF. A third order accurate scheme for variably saturated groundwater flow modelling. *Communications in Numerical Methods in Engineering* 2004; **20**(5):379–389.
19. 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**:1147–1163.
20. Ross PJ, Parlange J-Y. Comparing exact and numerical solutions of Richards' equation for one-dimensional infiltration and drainage. *Soil Science* 1994; **157**:341–344.
21. Milly PCD. A mass-conservative procedure for time stepping in models of unsaturated flow. *Advances in Water Resources* 1985; **8**:32–36.
22. van Genutchen MT. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal* 1980; **44**:892–898.
23. Carsel RF, Parrish RS. Developing joint probability distributions of soil water characteristics. *Water Resources Research* 1988; **24**:755–769.

