INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **35**: 25–38

Arnoldi and Crank–Nicolson methods for integration in time of the transport equation

Giorgio Pini* and Giuseppe Gambolati

Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, University of Padua, Padua, Italy

SUMMARY

A comparison is made between the Arnoldi reduction method and the Crank–Nicolson method for the integration in time of the advection–diffusion equation. This equation is first discretized in space by the classic finite element (FE) approach, leading to an unsymmetric first-order differential system, which is then solved by the aforementioned methods. Arnoldi reduces the native FE equations to a much smaller set to be efficiently integrated in the Arnoldi vector space by the Crank–Nicolson scheme, with the solution recovered back by a standard Rayleigh–Ritz procedure. Crank–Nicolson implements a time marching scheme directly on the original first-order differential system. The computational performance of both methods is investigated in two- and three-dimensional sample problems with a size up 30000. The results show that in advection-dominated problems less then 100 Arnoldi vectors generally suffice to give results with a 10^{-3}–10^{-4} difference relative to the direct Crank–Nicolson solution. However, while the CPU time with the Crank–Nicolson starts from zero and increases linearly with the number of time steps used in the simulation, the Arnoldi requires a large initial cost to generate the Arnoldi vectors with subsequently much less expensive dynamics for the time integration. The break-even point is problem-dependent at a number of time steps which may be for some problems up to one order of magnitude larger than the number of Arnoldi vectors. A serious limitation of Arnoldi is the requirement of linearity and time independence of the flow field. It is concluded that Arnoldi can be cheaper than Crank–Nicolson in very few instances, i.e. when the solution is needed for a large number of time values, say several hundreds or even 1000, depending on the problem. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: Arnoldi method; finite elements; iterative and direct methods; linear systems; transport equation

1. INTRODUCTION

Recently, reduction or spectral methods have received some attention for the solution of ordinary first-order differential equations of the form

* Correspondence to: Dipartimento di Metodi e Modelli Matematici, Università degli Studi di Padova, Via Belzoni 7, 35131 Padova, Italy.

Received July 1999
Revised March 2000

$$H\mathbf{c} + P\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t} + \mathbf{f} = 0 \tag{1}$$

arising from the finite element integration in space of the diffusion [1–4] or the advection–diffusion [5,6] equations. The basic idea underlying the reduction approach is to generate a basis of orthogonal vectors, called Lanczos [7] and Arnoldi [8] vectors, in a Krylov sub-space K_r to address diffusive and advective–diffusive type problems, respectively. Then the matrix $H^{-1}P$ is projected onto K_r by the Lanczos/Arnoldi vectors and the solution to Equation (1) is recovered following the standard Rayleigh–Ritz reduction procedure. A fundamental requirement for the Lanczos/Arnoldi method to be economically competitive with a more traditional time marching scheme for the integration of Equation (1) is that the size of the Lanczos/Arnoldi space be much smaller than the size n of Equation (1) and a cost-effective procedure be found to implement the Lanczos/Arnoldi recursive relationship, which involves the execution of the expensive product between H^{-1} and a vector. If the above conditions are satisfied, the time solution of the reduced system, e.g. by the use of the Crank–Nicolson scheme [9], is much less expensive than the solution of the original equations by the same method. However, a reduction method requires the initial cost to generate the Lanczos/Arnoldi vector space, with a size m, determined by the number of vectors needed to provide the solution to Equation (1) with the same accuracy as a direct time marching scheme, e.g. the second-order accurate Crank–Nicolson scheme. An important point to consider is the number of time values at which the solution to Equation (1) is sought. Since the Crank–Nicolson method does not require an initial cost, it goes without saying that a Lanczos/Arnoldi method can be superior to a Crank–Nicolson only after a sufficiently large number of time steps. To investigate the break-even point in advective–diffusive problems is one of the issues of the present paper. Another issue is finding out how many Arnoldi vectors prove necessary to ensure a solution with an accuracy comparable with the direct Crank–Nicolson solution, say 10^{-3}–10^{-4} in relative difference. One major observation to make is that a reduction method is less general and more difficult to implement on a computer than a standard time marching scheme and can offer a potentially worthwhile alternative only as long as Equation (1) is linear and H and P are time independent. This implies, for instance, that time-dependent velocity fields may seriously hamper the use of Arnoldi as they would reflect on a time dependent matrix H. Therefore, the comparison makes special sense and is carried out under the above mentioned restrictive assumptions. Actually, some authors (e.g. Reference [10]) have integrated Equation (1) by a Krylov approximation method with a forcing vector $\mathbf{f}$, which is time-dependent. Others (e.g. Reference [11]) have extended the approach to systems of non-linear differential equations using Krylov sub-space approximations to the exponential of the Jacobian. It must be observed, however, that these extensions require mathematically complex and computationally heavy algorithms, involving the product of a matrix exponential with a vector (based on some rational or polynomial approximation), which are far more difficult than the direct Crank–Nicolson solution in the reduced Krylov sub-space. Furthermore, to our knowledge, no techniques has been developed to address the problem, which may occur frequently in the applications, where either H or P or both are time-dependent (e.g. non-stationary flow field in compressible porous media).

This paper is organized as follows. First, the generation of the Arnoldi vectors and the Arnoldi reduction method are briefly reviewed. Then, four problems of increasing size ($n = 274$, 3321, 13041 and 29889) in two- and three-dimensional settings are solved using the Arnoldi method and the Crank–Nicolson scheme. The results are compared with an estimate of the proper size m of the Arnoldi space and the determination of the break-even point for each example. Diffusion- and convection-dominated problems with Peclet numbers of 1, 5 and 10 are considered. Finally, a set of conclusive remarks on the relative performance of the two time integration approaches and some recommendations on their use are issued.

2. CRANK–NICOLSON AND ARNOLDI REDUCTION METHODS

2.1. Crank–Nicolson method

If H and P are both time -independent matrices, the integration in time of Equation (1) by the second-order accurate Crank–Nicolson scheme [9] yields the following algebraic system to be solved repeatedly in time:

$$\left(H + \frac{2P}{\Delta t}\right)\mathbf{c}_{t+\Delta t} = \left(\frac{2P}{\Delta t} - H\right)\mathbf{c}_t - \mathbf{f}_t - \mathbf{f}_{t+\Delta t} \tag{2}$$

The solution to Equation (2) can be addressed by a direct elimination method [12] after an optimal preliminary reordering or, more conveniently, if the size of Equation (2) is large, by an iterative method. For instance, a projection method based on the unsymmetric conjugate gradients [13].

2.2. Arnoldi method

Given the Krylov space $K_r = \text{span}\{\mathbf{v}_1, A\mathbf{v}_1, \ldots, A^{m-1}\mathbf{v}_1\}$, with A an unsymmetric $n \times n$ matrix and $\mathbf{v}_1$ an arbitrary unit vector, the Arnoldi recursive procedure generates an orthogonal vector basis in K_r. At each step the algorithm [8] requires the product between the Arnoldi vector $\mathbf{v}_j$ and A to get $\mathbf{w}_j$, which is orthogonalized against all the previous $\mathbf{v}_i$ by a standard Gram–Schmidt procedure. Hence

$$\mathbf{w}_j = \eta_{j+1,j}\mathbf{v}_{j+1} = A\mathbf{v}_j - \sum_{i=1}^{j} \eta_{ij}\mathbf{v}_i \tag{3}$$

where $\eta_{ij} = \mathbf{v}_i^T A\mathbf{v}_j$ and $\eta_{j+1,j} = \|\mathbf{w}_j\|$. If $V_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ is an $n \times m$ matrix, with its columns as the first m Arnoldi vectors, K_m is an $m \times m$ Hessemberg matrix, with the non-zero elements equal to η_{ij} and $I_m = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m]$ is the identity matrix, we have [14]

$$AV_m = V_m K_m + \mathbf{w}_m\mathbf{e}_m^T \tag{4}$$

$$V_m^T V_m = I_m, \qquad V_m^T\mathbf{w}_m = 0 \tag{5}$$

$$V_m^T A V_m = K_m \tag{6}$$

Setting $A = H^{-1}P$, Equation (6) takes on the form

$$V_m^T H^{-1} P V_m = K_m \tag{7}$$

Applying the Ritz–Rayleigh reduction process to Equation (1), which is left-multiplied by $V_m^T H^{-1}$, yields

$$V_m^T \mathbf{c} + V_m^T H^{-1} P \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t} + V_m^T H^{-1} \mathbf{f} = 0 \tag{8}$$

Solution $\mathbf{c}$ is now expressed in the Arnoldi vector space

$$\mathbf{c} = V_m \mathbf{u} \tag{9}$$

Equation (8) turns into

$$V_m^T V_m \mathbf{u} + V_m^T H^{-1} P V_m \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + V_m^T H^{-1} \mathbf{f} = 0 \tag{10}$$

By the use of Equations (5) and (7), with $\mathbf{g} = V_m^T H^{-1} \mathbf{f}$, we arrive at a reduced system of size m

$$\mathbf{u} + K_m \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{g} = 0 \tag{11}$$

which may be integrated in time by the Crank–Nicolson scheme (2). The solution is then back-recovered by Equation (9).

The Arnoldi algorithm (3) applied to the matrix $A = H^{-1}P$ for $j = 1, 2, \ldots, m$ takes on the following expression:

$$\mathbf{w}_j = \eta_{j+1,j} \mathbf{v}_{j+1} = H^{-1} P \mathbf{v}_j - \sum_{i=1}^{j} \eta_{ij} \mathbf{v}_i \tag{12}$$

where $\eta_{ij} = \mathbf{v}_i^T H^{-1} P \mathbf{v}_j$ and $\eta_{j+1,j} = \|\mathbf{w}_j\|$.

A lot of research has been done to address the solution to Equation (1) with $P = I$ using the exponential of the matrix H projected onto the Krylov sub-space K_r with respect to the Arnoldi basis V_m; see, for example, References [10,15]. With this approach, the product between $\exp(-tH)$ and a vector $\mathbf{v}$, which is theoretically required in the formally exact integration of Equation (1), is approximated as

$$\exp(-tH)\mathbf{v} \simeq \beta V_m \exp(-tK_m)\mathbf{e}_1 \tag{13}$$

where $\mathbf{e}_1$ is the unit vector of size m with its first component equal to one, β is the Euclidean norm of $\mathbf{v}$ and K_m is the Hessemberg matrix obtained from the above Arnoldi process, with $A = H$ (and $P = I$). Most of the improvements in the matrix exponential approach (see Reference [16] for a review) concern the efficient calculation of Equation (13), for instance, using suitable approximate expressions for the exponential (e.g. References [17,18]). We observe that the efficient computation of Equation (13) is not an issue of the present study. In fact, as will be demonstrated later, the major burden in the Arnoldi reduction method is not the efficient correct calculation of Equation (13) or the accurate solution to the equivalent system (11), but the generation of the Arnoldi vector space, which requires a computational cost much larger than the cost of solving the reduced problem in the Krylov sub-space. Therefore, we have elected to solve Equation (11) by a traditional Crank–Nicolson scheme with an appropriate time step Δt, even though we are aware that using the matrix exponential approach a larger, or much larger, Δt can be used provided that V_m is sufficiently large (see, for instance, Reference [10], p. 1254). Saving time in the solution of Equation (11) will not help to substantially abate the cost of the Arnoldi method.

Another issue to point out is that the above Arnoldi algorithm uses H^{-1} instead of H, as is done in the papers mentioned previously. Using H or H^{-1} produces different Arnoldi vectors and hence a different approximate solution in the respective space. In other words, convergence to the correct solution may depend on whether H or H^{-1} is implemented into the Arnoldi recursive equations. We do not use H, as a much larger Arnoldi space is needed to converge at intermediate/large time values, as will be shown later. Using H^{-1} requires a higher computational cost to perform the product $H^{-1}P\mathbf{q}_j$ in Equation (12). However, this larger cost is fully counterbalanced by the faster convergence in the corresponding Krylov space (see Section 6).

3. DESCRIPTION OF THE SAMPLE PROBLEMS

3.1. Regular two-dimensional grid

Consider a rectangular domain 1×0.5 m with Dirichlet conditions prescribed along $x = 0$, namely $c(0, t) = 1$ for $0.2 \leq y \leq 0.3$ and $c(0, t) = 0$ elsewhere. The Neumann condition $\partial c/\partial n = 0$ is prescribed on the remaining sides. For the Darcy velocity, we assume $v_x = 1$ m s^{-1}, $v_y = 0$ m s^{-1} and the time integration step $\Delta t = 0.02$ s. We will explore five sample problems in relation to five different uniform triangular discretizations of the domain and Peclet numbers $Pe = \Delta x/\alpha_{\mathrm{L}}$, with α_{L} as the longitudinal (along x) dispersivity and the transverse dispersivity $\alpha_{\mathrm{T}} = \alpha_{\mathrm{L}}$. Table I shows the spacings $\Delta x = \Delta y$, the size N of the grids, the total number of triangular elements, NE, α_{L} and Pe for the various cases (1a–2b).

3.2. Regular three-dimensional grid

The domain is a box $1 \times 0.5 \times 1$ m; $v_x = 1$ m s^{-1}; $v_y = v_z = 0$ m s^{-1}, $\Delta t = 0.02$ s. The boundary conditions are obtained by extending those of the previous example along the z-direction. The vertical side is discretized using a uniform spacing $\Delta z = 0.125$ m, with tetrahedral elements generated as is described in Reference [19], starting from the two-dimensional mesh of the previous problem. The data of Table I are used for this problem as well (cases 3a and 3b).

Table I. Nodal spacing $\Delta x = \Delta y$, number of nodes n, number of finite elements (triangles in the two-dimensional (2D) problem, tetrahedrons in the three-dimensional (3D) problem) NE, dispersivity α_L and α_T, and the maximum Peclet number (Pe) for the various test cases.

Problem	$\Delta x = \Delta y$ (m)	n	NE	$\alpha_L = \alpha_T$ (m)	Pe
1a (2D)	0.0125	3321	6400	0.0125	1
1b (2D)	0.0125	3321	6400	0.0025	5
1c (2D)	0.0125	3321	6400	0.000125	10
2a (2D)	0.00625	13 041	25 600	0.00625	1
2b (2D)	0.00625	13 041	25 600	0.00125	5
3a (3D)	0.0125	29 889	153 600	0.0125	1
3b (3D)	0.0125	29 889	153 600	0.0025	5
4a (2D)		274	533	6000	1
4b (2D)		274	533	1200	5

3.3. Irregular two-dimensional grid

The domain here is a circle with the centre at the origin and radius $r = R = 10000$ m (Figure 1). We assume $c = 1$ at $r = 0$ and a radial velocity over a triangle equal to

$$v_r = \frac{Q}{2\pi r} \Rightarrow v_x = \frac{Qx_G}{2\pi r^2}, \qquad v_y = \frac{Qy_G}{2\pi r^2}, \quad Q = 1$$

with (x_G, y_G) co-ordinates of the centre of gravity of each triangle. The triangular mesh of Figure 1 has been generated using the algorithm given in Reference [20]. Table I summarizes

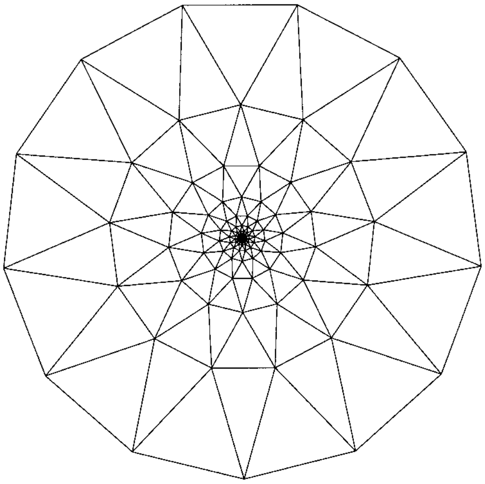

Figure 1. Irregular two-dimensional grid, $n = 274$, NE $= 533$.

the characteristic information for the cases 4a and 4b in this example, where $\Delta t = 84600$ s is assumed.

4. SOLUTION OF LINEAR SYSTEMS

The algebraic linear systems arising from the Crank–Nicolson integration of both Equations (1) and (2) are solved by the use of either the iterative projection method, bi-conjugate gradient stabilized (Bi-CGSTAB) method [13] or a direct elimination scheme [12]. More precisely, system (2), whose coefficient matrix is $H + 2P/\Delta t$, is usually solved by Bi-CGSTAB preconditioned with the incomplete triangular factor and no fill-in (ILU) [21–23]. Bi-CGSTAB requires on the average ten iterations per time step. The only exception is problem 2b, for which Bi-CGSTAB preconditioned with ILU does not converge. In this case, a direct method is used instead. With reference to the computation of the m Arnoldi vectors, Equation (12), setting $\bar{\mathbf{q}}_j = P\mathbf{q}_j$, the products $H^{-1}\bar{\mathbf{q}}_j = \mathbf{z}_j$ are performed implicitly by solving the linear system

$$H\mathbf{z}_j = \bar{\mathbf{q}}_j \tag{14}$$

usually by a direct approach. The only exception is problem 3b, for which the direct approach turned out to be unfeasible. This is so because the memory requirement of the direct solver for problem 3b exceeded the memory capacity of our workstation (256 Mbytes), despite the use of an optimal reordering algorithm with a substantial reduction of the fill-in in the triangular factors. It is to be noted, however, that Equation (14) for problem 3b could not be solved successfully by Bi-CGSTAB either, as it also failed to converge. This example shows that in three-dimensional particularly difficult problems (e.g. large Peclet number), the Arnoldi solution strategy can hardly be applied.

It is worth noting that the Arnoldi method requires the solution of m linear systems of the type (14), where the coefficient matrix H is the stiffness–advection matrix arising from the FE discretization of the transport equation. Compared with the corresponding system (2), Equation (14) is more difficult to solve since H has a conditioning that is surely worse than that of $H + 2P/\Delta t$. This is due to the fact that the capacity matrix P is symmetric positive definite and hence enhances the conditioning of H.

5. COMPARISON OF ARNOLDI AND CRANK–NICOLSON METHODS

Figure 2 shows the behaviour of the maximum difference *errmax* between the solutions to Equation (1), obtained with the direct integration by the Crank–Nicolson method, and the use of the Arnoldi approach versus the number, m, of Arnoldi vectors. If $c_i^{(C)}$and $c_i^{(A)}$ denote the Crank–Nicolson and Arnoldi solutions respectively, we define

$$e_l = \max\{|c_i^{(C)} - c_i^{(A)}|\colon i = 1, \ldots, n\}$$

$$errmax = \max\{e_l\colon l = 1, \ldots, 50\}$$

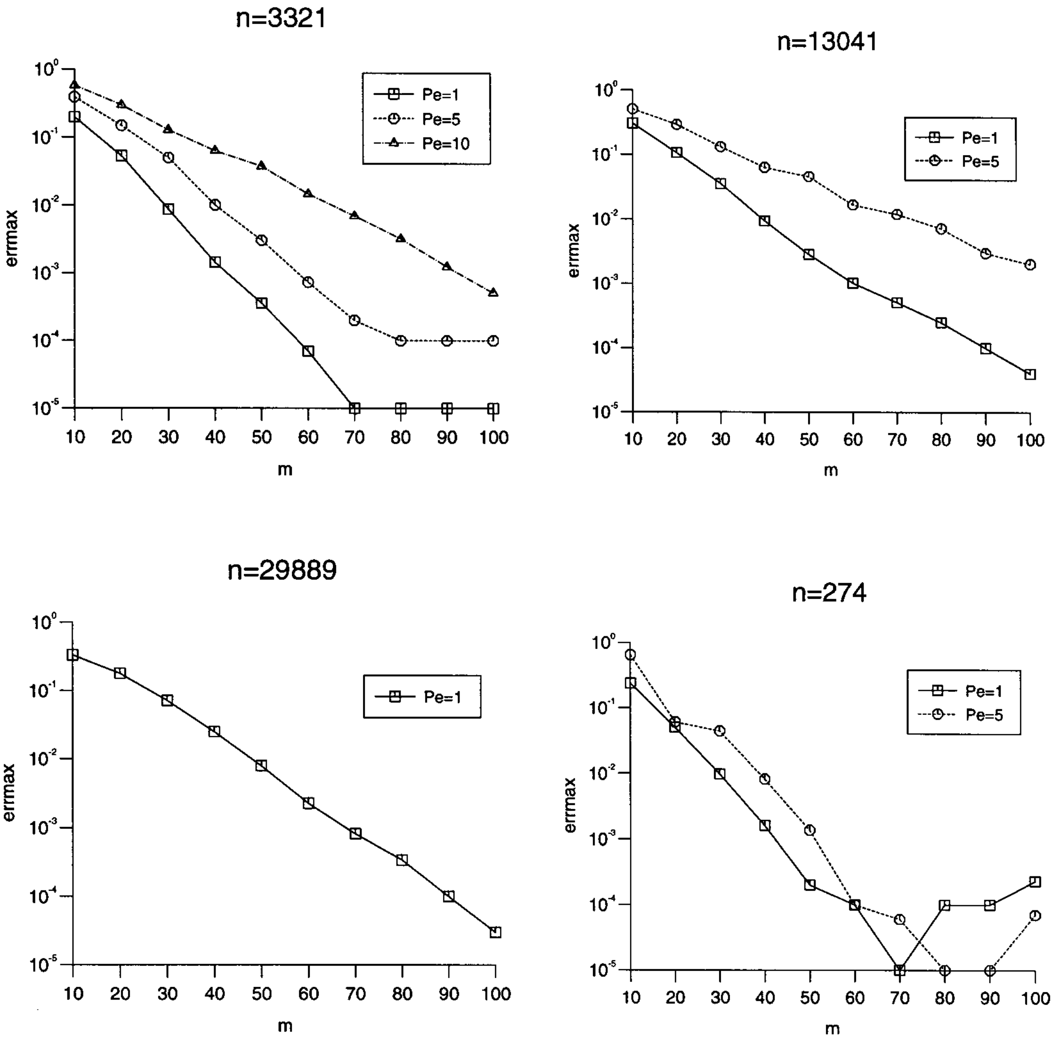

Figure 2. Behaviour of *errmax* versus the number m of Arnoldi vectors.

where 50 is the number of time steps used in the integration of Equation (1). All the problems reported in Table I are represented in Figure 2, except for test case 3b, for which neither the direct nor the iterative approaches are successful in the solution of system (14). Inspection of this figure indicates that a number of Arnoldi vectors m smaller than 100 suffices to provide *errmax* between 10^{-3} and 10^{-4}, i.e. small enough for $c_i^{(A)}$ to be considered a good approximation of $c_i^{(C)}$, irrespective of the problem size n. Note that m tends to increase for increasing Pe. This would suggest that for large Peclet numbers, i.e. in highly convection-dominated problems, the number of Arnoldi vectors needed to obtain an accurate solution (or at least results close to the Crank–Nicolson ones) may exceed 100.

We now try to compare the CPU times required by both approaches and find out the break-even point where they are equally expensive. This point is given by the number of time steps used in the simulation beyond which the Arnoldi method is economically more convenient than Crank–Nicolson. To perform a meaningful analysis that proves to be general as much as possible, we show the results for $m = 30$, 40, 50 and 100, and use the most robust solver for either approach, i.e. usually the Bi-CGSTAB solver for direct Crank–Nicolson and the Gauss elimination solver for Arnoldi (remember that Bi-CGSTAB may not converge when solving (14)). Figure 3 shows the results for problems 1a, 1b and 1c. Note that for any given m Arnoldi times are almost insensitive to Pe and to the number M of times steps with most of the cost spent to produce the Arnoldi vectors. By contrast, as expected, Crank–Nicolson cost increases linearly with M and the straightline slope also increases with Pe since a larger Pe can make Bi-CGSTAB more slowly convergent. The representative break-even point for $m = 100$ in these three figures is around $M = 300$. In Figure 4 (problem 2a) the break-even point is beyond $M = 1500$. A somewhat anomalous result is that of Figure 4 (problem 2b). Here it proved necessary to solve (2) by a direct solver since Bi-CGSTAB does not converge. Of course, the break-even point here is much smaller, of the order of 100, as Crank–Nicolson is particularly expensive in this case. Figure 3 (problem 3a) points out again an M of the order of 1000 or more, although in Arnoldi as well we elected to use Bi-CGCTAB as it proves more efficient than the direct solver. The results for problem 3b are not given since for this case Arnoldi could not be addressed on our workstation, as was mentioned before. For small problems 4a and 4b, M is also small, of the order of 100 or smaller, as fewer Arnoldi vectors are required to capture the solution (Figure 4).

6. USING $P^{-1}H$ TO BUILD THE KRYLOV SUB-SPACE

Some people might think of solving system (1) in the form

$$P^{-1}H\mathbf{c} + \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t} + P^{-1}\mathbf{f} = 0 \tag{15}$$

and generating the Arnoldi vector sequence by the use of matrix $P^{-1}H$ [10]. In such a way the Arnoldi recursive scheme would become cheaper since it would require the solution of linear systems with matrix P, which is symmetric positive definite, in place of matrix H, which is unsymmetric indefinite. We will call this procedure Arnoldi 2 as opposed to the Arnoldi 1 procedure described in Section 2.2. At a first glance, Arnoldi 2 appears to be the classical Colombo's egg. Unfortunately, it does not work as is demonstrated in Table II, showing *errmax* versus the Arnoldi 2 space size m over 50 time steps for the problems discussed earlier and comparing the Arnoldi 1 and 2 results. We note from Table II that

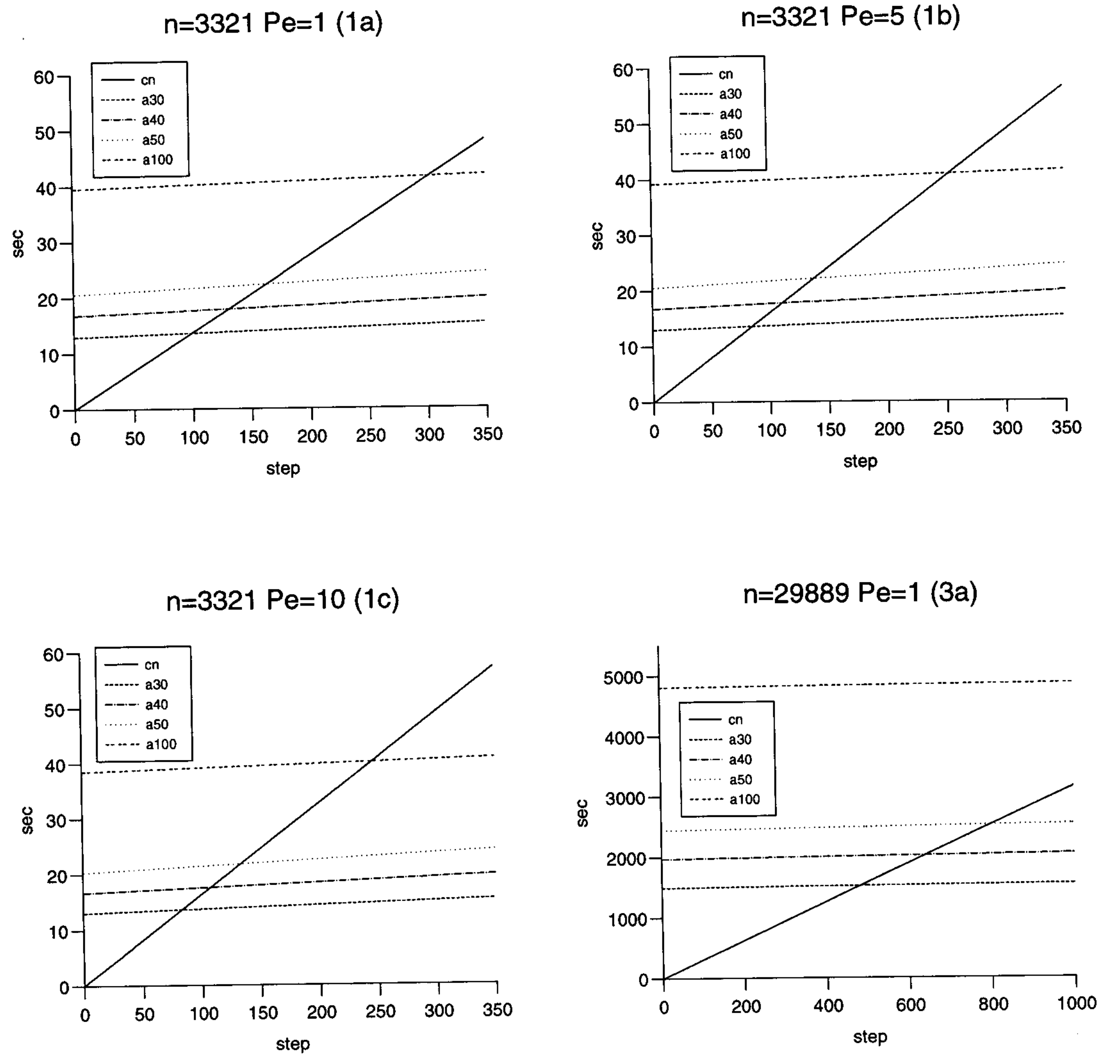

Figure 3. CPU time versus the time steps for problems 1a, 1b, 1c, 3a.

convergence in the Arnoldi 2 space is very slow with a much larger Arnoldi basis needed to achieve the same accuracy as with the original Arnoldi 1 sequence.

This outcome comes as no surprise, since Arnoldi (as well as Lanczos) converges earlier to the rightmost eigenpairs of the matrix used to form the Krylov sub-space. The solution to Equation (1) is essentially controlled by the leftmost eigenvectors of $H^{-1}P$ (for a

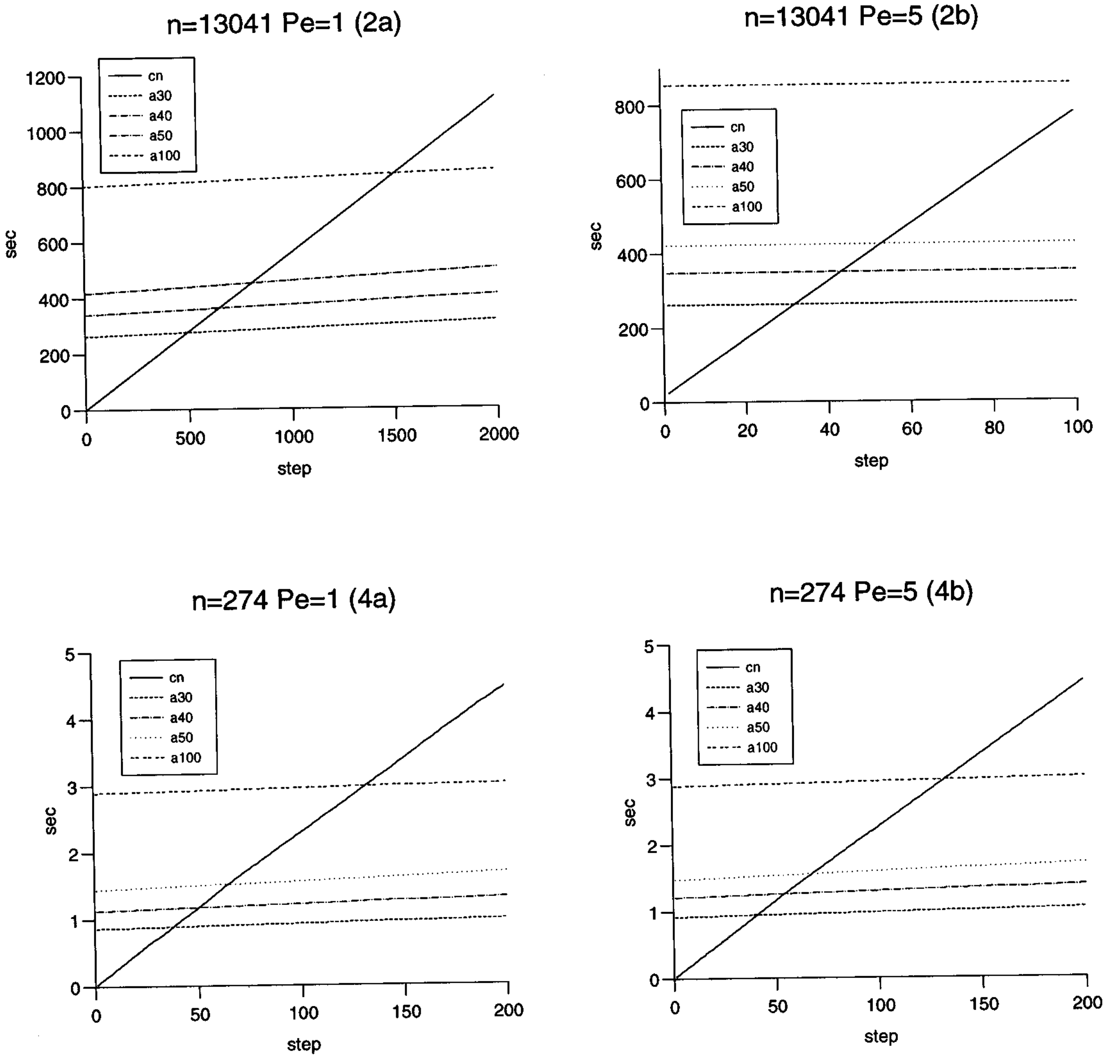

Figure 4. CPU time versus the time steps for problems 2a, 2b, 4a, 4b.

time-independent source vector **f** as was the case in the examples discussed above), and this accounts for the faster convergence of the Arnoldi 1 method, especially at intermediate/large values of time.

Finally, it should be mentioned that not only Arnoldi 2 delays the convergence to the desired solution but it also requires increasing the core memory to store the larger number of Arnoldi 2 vectors generated by the procedure. So Arnoldi 2 must be dismissed on the ground of both computational inefficiency and impracticality of for large size problems.

7. CONCLUSIONS

An analysis has been performed and a comparison has been made between the computational behaviour and cost of the Crank–Nicolson and Arnoldi methods for the integration in time of linear differential systems, which arise from the FE solution of the sub-surface transport equation. The following results are worth summarizing:

1. The Arnoldi method requires the *a priori* generation of an Arnoldi vector space with a size *m*, which has been found to be of the order of 100 also in large advective-dominated problems with many thousands nodes.
2. The cost of the Arnoldi method is basically determined by the cost for generating the Arnoldi vector space. Time integration in this space is relatively inexpensive. Therefore, any effort to improve the integration of the projected equations by the matrix exponential approach which may allow for a larger, or much larger, time step than the standard Crank–Nicolson scheme does not help substantially the Arnoldi method. Its basis cost is

Table II. *Errmax* versus size *m* of Arnoldi 1 and Arnoldi 2 vector spaces for the problems discussed earlier ($Pe = 1$).

m	$n = 274$		$n = 3321$	
	Arnoldi 1	Arnoldi 2	Arnoldi 1	Arnoldi 2
10	0.23693E+00	0.14457E+01	0.19724E+00	0.10607E+01
20	0.50140E−01	0.10119E+01	0.53510E−01	0.94917E+00
30	0.97200E−02	0.10119E+01	0.87500E−02	0.87057E+00
40	0.16100E−02	0.10119E+01	0.14400E−02	0.79927E+00
50	0.20000E−03	0.10120E+01	0.36000E−03	0.74271E+00
60	0.10000E−03	0.10121E+01	0.70000E−04	0.69581E+00
70	0.10000E−04	0.10126E+01	0.10000E−04	0.65040E+00
80	0.10000E−03	0.10136E+01	0.10000E−04	0.59361E+00
90	0.10000E−03	0.10164E+01	0.10000E−04	0.55925E+00
100	0.23000E−03	0.10332E+01	0.10000E−04	0.52041E+00
150	—	0.80880E+00		0.31135E+00
	$n = 13\,041$		$n = 29\,889$	
10	0.30412E+00	0.11254E+01	0.33118E+00	0.11381E+01
20	0.10707E+00	0.11088E+01	0.17860E+00	0.10634E+01
30	0.35290E−01	0.10935E+01	0.71540E−01	0.10103E+01
40	0.93200E−02	0.10533E+01	0.25110E−01	0.96211E+00
50	0.28100E−02	0.10308E+01	0.79700E−02	0.85831E+00
60	0.10200E−02	0.10138E+01	0.22700E−02	0.83236E+00
70	0.51000E−03	0.98186E+00	0.82000E−03	0.77328E+00
80	0.25000E−03	0.95311E+00	0.34000E−03	0.74761E+00
90	0.10000E−03	0.91499E+00	0.10000E−03	0.69283E+00
100	0.40000E−04	0.89872E+00	0.30000E−04	0.66289E+00
150	—	0.78985E+00		0.47301E+00

produced by the repeated solution of m linear systems with the stiffness–advection unsymmetric matrix H, which is generally poorly conditioned and hence not recommended for the use of iterative solvers.

3. The Crank–Nilcolson method does not have an initial cost and displays a cost dynamics, which grows linearly with the number M of time steps. Since the Crank–Nicolson matrix $H + 2P/\Delta t$ is better conditioned than H because of the addition of the symmetric positive definite matrix $2P/\Delta t$, an efficient conjugate gradient type solver, such as Bi-CGSTAB, can be generally used. This choice, however, makes the solver performance, i.e. the slope of the cost straight line versus M, sensitive to some extent to the grid Peclet number, namely the cost is higher for advection-dominated problems. By distinction, Arnoldi is not sensitive to the Peclet number, at least up to the Peclet values explored by the present study.
4. The break-even point of the Crank–Nicolson and Arnoldi methods for large problems, i.e. with a number of nodes one order of magnitude or more larger than size of the Arnoldi vector space, can be several hundreds up to more than 1000 for the most difficult problems.
5. For very large problems where advection dominates, it may prove necessary to use a direct solver for Crank–Nicolson as well. In such a case, the performance of Arnoldi relative to Crank–Nicolson increases significantly and the break-even point can be substantially lower.
6. In some very large problems dominated by advection a direct solution approach may be unfeasible for Arnoldi and Bi-CGSTAB may not converge, hence Arnoldi cannot be used.
7. Generation of the Arnoldi vector space by the use of $P^{-1}H$ instead of $H^{-1}P$ would reduce the cost required by each single Arnoldi vector but would greatly slow down the convergence to the solution in the Krylov sub-space and create the problem of preserving in the core memory a large number of vectors. Therefore, it is not a viable alternative to the Arnoldi procedure described in Section 2.2.

The general conclusive remark from the previous analysis is that the Crank–Nicolson method is more robust and less expensive than Arnoldi for medium to large size transport problems. The break-even point occurs after a large number of time steps, typically several hundreds or even 1000.

ACKNOWLEDGMENTS

This work has been partially supported by the Italian MURST project 'Numerical methods for large scale problems in science and engineering' and by INCO-COPERNICUS program of the European Community (Contract ERBIC15CT960211).

REFERENCES

1. Sahuquillo A. An eigenvalue numerical technique for solving unsteady linear groundwater models continuously in time. *Water Resources and Research* 1983; **19**: 87–93.
2. Dunbar WS, Woodbury AD. Application of the Lanczos algorithm to the solution of the groundwater flow equation. *Water Resources and Research* 1989; **25**: 551–558.
3. Gambolati G. On time integration of groundwater flow equations by spectral methods. *Water Resources and Research* 1993; **29**: 1257–1267.

4. Dunbar WS, Woodbury AD, Nour-Omid B. Comment on 'On time integration of groundwater flow equations by spectral methods' by G. Gambolati. *Water Resources and Research* 1994; **30**: 2347–2355.
5. Hwang JC, Cho WC, Yeh GT. An eigenvalue solution continuous in time to spatially discretized solute transport equation in steady groundwater flow. *Water Resources and Research* 1984; **20**: 1725–1732.
6. Nour-Omid B, Dunbar WS, Woodbury AD. Lanczos and Arnoldi methods for the solution of convection–diffusion equations. *Computing Methods in Applied Mechanics and Engineering* 1991; **88**: 75–95.
7. Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards* 1950; **45**: 255–282.
8. Arnoldi WE. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly in Applied Mathematics* 1951; **9**: 17–29.
9. Crank J, Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proceedings of the Cambridge Philosophical Society* 1947; **40**: 50–67.
10. Gallopoulos E, Saad Y. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM Journal of Scientific and Statistical Computing* 1992; **13**: 1236–1264.
11. Hockbruck M, Lubich C, Selhofer H. Exponential integrators for large systems of differential equations. *SIAM Journal of Scientific Computing* 1998; **19**: 1552–1574.
12. Duff IS, Erisman AM, Reid JK. *Direct Methods for Sparse Matrices*. Oxford University Press (Clarendon): Oxford, 1986.
13. van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing* 1992; **13**: 631–644.
14. Saad Y. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press: Manchester, 1992.
15. Hochbruck M, Lubich C. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal of Numerical Analysis* 1997; **34**: 1911–1925.
16. Moler C, Loan CV. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Reviews* 1978; **20**: 801–836.
17. Saad Y. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal of Numerical Analysis* 1992; **29**: 209–228.
18. Bergamaschi L, Vianello M. Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numerical Linear Algebra with Applications* 2000; **7**: 27–45.
19. Gambolati G, Pini G, Tucciarelli T. A 3D finite element conjugate gradient model of subsurface flow with automatic mesh generation. *Advances in Water Resources* 1986; **3**: 34–41.
20. Gambolati G. *Ottimizzazione di reticoli triangolari per il metodo agli elementi finiti in idrologia*. Atti convegno AICA: Milan, 1972.
21. Kershaw DS. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics* 1978; **26**: 43–65.
22. Meijerink JA, van der Vorst HA. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics in Computing* 1977; **31**: 148–162.
23. Pini G, Putti M. Krylov methods in the finite element solution of groundwater transport problems. In *Computational Methods in Water Resources X*, vol. 1, Peters A, Wittum G, Herrling B, Meissner V, Brebbia CA, Gray WG, Pinder GF (eds). Kluwer Academic: Dordrecht, 1994; 1431–1438.