NON-SYMMETRIC CG-LIKE SCHEMES AND THE FINITE ELEMENT SOLUTION OF THE ADVECTION–DISPERSION EQUATION

ALEXANDER PETERS

IBM Germany Scientific Center, Vangerowstr. 18, D-6900 Heidelberg, Germany

SUMMARY

Seven leading iterative methods for non-symmetric linear systems (GMRES, BCG, QMR, CGS, Bi-CGSTAB, TFQMR and CGNR) are compared in the specific context of solving the advection–dispersion equation by a classic approach: The space derivatives are approximated by linear finite elements while an implicit scheme is used to integrate the time derivatives. Convergence formulas that predict the behaviour of the iterative methods as a function of the discretization parameters are developed and validated by experiments. It is shown that all methods converge nicely when the coefficent matrix of the linear system is close to normal and the finite element approximation of the advection–dispersion equation yields accurate results.

KEY WORDS Advection–dispersion equation Finite elements Iterative methods Non-symmetric linear systems Conjugate gradients

INTRODUCTION

In this paper we compare seven leading iterative methods for non-symmetric systems of linear equations in the context of solving the advection–dispersion equation

$$\frac{\partial c}{\partial t}+\frac{\partial}{\partial x_i}(v_i c)-\frac{\partial}{\partial x_i}\left(d_{ij}\frac{\partial c}{\partial x_j}\right)=0. \tag{1}$$

Equation (1) is defined over a domain $G\subset\mathbb{R}^3$ with initial conditions and Dirichlet and Cauchy conditions on the boundary ∂G. The unknown function c represents the solute concentration, t and x_i are the time and spatial co-ordinates, v_i is the velocity vector, and d_{ij} the dispersion tensor, defined as the sum of molecular diffusion and mechanical dispersion.[1]

One of the most popular approaches for the solution of equation (1) consists of the finite element approximation of the space derivatives and a weighted finite difference scheme for the integration of the time derivatives. This approach requires the solution of a large but sparse system of linear equations in each iteration. Depending upon the weighting factors chosen for the time integration, the coefficient matrix of the linear system is either Symmetric and Positive Definite (SPD) or non-symmetric.

The 'classical' Conjugate Gradient (CG) method[2] with preconditioning is one of the most powerful algorithms for the solution of sparse SPD linear systems. Eiermann *et al.*[3] showed that the combination of the symmetric semi-implicit time integration scheme of Leismann and Frind[4]

0271-2091/93/230955-20$15.00

Received 20 January 1993
Revised 22 July 1993

and the classical CG method represents a robust and very efficient strategy for the solution of the advection–dispersion equation.

We now return to a more popular class of non-symmetric schemes. In particular, we employ the Crank–Nicolson scheme and linear finite elements to approximate equation (1) by a sequence of non-symmetric linear systems. Seven leading CG-like methods are applied to the solution of these systems. Besides the well-known algorithms CGNR,[2] BCG[5,6] and GMRES,[7] we implemented several schemes published during the last years: CGS,[8] Bi-CGSTAB,[9] QMR[10,11] and TFQMR.[12] We recall that excellent review papers presenting these methods have been published by Saad,[13] van der Vorst[14] and Freund *et al.*[15] Nachtigal *et al.*[16] constructed a set of simple artificial examples to illustrate the significant differences in capabilities between CGNR, GMRES and CGS.

The outline of this paper is as follows. In Section 2 we review key issues related to the convergence properties of CG-like schemes. The discretization techniques and a set of model problems used to generate test linear systems from the advection–dispersion equation are briefly presented in Section 3. In Section 4 we apply GMRES and CGNR to the solution of a one-dimensional model problem and estimate the convergence rates as a function of the discretization parameters. Additionally, we look for those parameters that produce solutions of similar accuracy with minimal computational effort. The theoretical results are validated by numerical experiments performed with all considered CG-like methods in Section 5.

2. CG-LIKE ALGORITHMS

We apply the algorithms GMRES, BCG, QMR, CGS, Bi-CGSTAB, TFQMR and CGNR to the solution of the system of linear equations

$$A\mathbf{c}=\mathbf{b}, \tag{2}$$

where $A\in\mathbb{R}^{N\times N}$ and $\mathbf{c},\mathbf{b}\in\mathbb{R}^N$. The considered algorithms are extensions of the classical CG-method[2] to non-symmetric systems.

CG-like algorithms seek the kth approximate solution $\mathbf{c}_k$ to equation (2) from the subspace $\mathbf{c}_0+K_k(\mathbf{r}_0, A)$, where $K_k(\mathbf{r}_0, A)\equiv\operatorname{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}$ is the kth Krylov subspace generated by the coefficient matrix A and the initial residual vector $\mathbf{r}_0=\mathbf{b}-A\mathbf{c}_0$. The residual vector of the kth step $\mathbf{r}_k=\mathbf{b}-A\mathbf{c}_k$ can be written as

$$\mathbf{r}_k=\mathbf{r}_0-Aq_{k-1}(A)\mathbf{r}_0=p_k(A)\mathbf{r}_0, \tag{3}$$

where $q_{k-1}\in\prod_{k-1}$ is a polynomial of degree at most $k-1$ and $p_k(z)=1-zq_{k-1}(z)\in\prod_k$ with $p_k(0)=1$. $\prod_k$ denotes the set of all polynomials of degree at most k.

Faber and Manteuffel[17] demonstrated that with a few exceptions, CG-like schemes cannot satisfy the following properties simultaneously:

(a) construct iterates $\mathbf{c}_k$ that minimize the residual vector in some norm, i.e. $\|\mathbf{r}_k\|_2\equiv\sqrt{(\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k)}$ is minimal for all $\mathbf{c}_k\in\mathbf{c}_0+K_k(\mathbf{r}_0, A)$,

(b) find $\mathbf{c}_k$ by short recurrence relations such that the work and storage requirements remain small and constant while the iterative process moves forward. (Note that the classical CG scheme for SPD matrices is the most remarkable exception from this rule).

We subdivide the implemented CG-like schemes into four groups according to their convergence properties:

2.1. *Residual minimization schemes (GMRES)*

GMRES[7] constructs iterates that satisfy the residual minimization property

$$\|\mathbf{r}_k\|_2 = \min_{p \in \prod_k,\, p(0)=1} \|p(A)\mathbf{r}_0\|_2 . \tag{4}$$

The approximate solution $\mathbf{c}_k$ is computed by first creating an orthonormal basis for the Krylov subspace $K_k(\mathbf{r}_0, A)$ and then solving a least-squares problem to determine the residual that satisfies equation (4). Saad and Schultz[7] showed that the least-squares problem has unique solution and consequently GMRES cannot break down. Moreover, since the residual norm is minimized in each step, it cannot increase while the iteration proceeds.

The most important drawback of the GMRES iteration stems from the fact that the Arnoldi orthogonalization procedure requires the storage of k vectors and the execution of $O(k)$ vector operations in each step. This makes the GMRES scheme very expensive when k increases. To limit the work and storage, Saad and Schultz[7] proposed the truncated version GMRES(m) that restarts the original algorithm after m steps, where m is some fixed parameter. Similar to the generic algorithm, the restarted version does not break down. However, the appropriate selection of m often requires additional experiments since the iterative process may begin to converge only after m had reached a certain value.

Equation (4) indicates that the convergence speed of GMRES depends upon the residual polynomial $p(A)$, i.e. $\|\mathbf{r}_k\|_2$ shrinks fast when $\|p(A)\|_2$ is small. Assume that matrix A is diagonalizable, i.e. there exists an invertible matrix $T \in \mathbb{R}^{N \times N}$ such that $T^{-1}AT$ has diagonal form. Saad and Schultz[7] showed that the convergence rate of GMRES is bounded above by

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq \kappa(T) \min_{p \in \prod_k,\, p(0)=1} \max_{\lambda_i \in \Lambda} |p(\lambda_i)|, \tag{5}$$

where $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ is the spectrum of A and $\kappa(T) = \|T\|_2 \, \|T^{-1}\|_2$ is the condition number of any matrix T of eigenvectors of A.

In practice, the use of equation (5) is limited to few special problems since neither the solution of the complex approximation problem of minimizing $\max_{\lambda_i \in \Lambda} |p(\lambda_i)|$ nor the estimation of $\kappa(T)$ are trivial. Moreover, if A is not normal and $\kappa(T)$ is large, any attempt to estimate the convergence of GMRES based on eigenvalues estimates may have little or no practical significance.[18] This assertion is sustained by the examples presented in Section 4.

2.2. *Lanczos-type schemes (BCG, QMR)*

The Bi-Conjugate Gradient (BCG) method[5,6] and the Quasi-Minimal Residual (QMR) method[10] seek the kth approximate solution $\mathbf{c}_k$ to the linear system of equation (2) from the same Krylov subspace $\mathbf{c}_0 + K_k(\mathbf{r}_0, A)$ as GMRES. By contrast to GMRES however, the 2-norm of the residuals computed by BCG and QMR do not satisfy the minimization property given by equation (4). Moreover, BCG and QMR employ the non-symmetric Lanczos process to generate basis vectors for the Krylov subspace, instead of the Arnoldi orthogonalization procedure used by GMRES.

The non-symmetric Lanczos process reduces matrix A to a tridiagonal form. Moreover, it generates bi-orthogonal basis vectors for the Krylov subspaces using only a pair of three-term recurrences (for details see Golub and van Loan[19]). Consequently, the Lanczos process can be completed with little work and storage per iteration. An important drawback of the Lanczos process is that it can breakdown before an invariant subspace is found for A. In most cases such unwanted terminations can be avoided with the help of so called 'look-ahead' steps.[20]

BCG generates the same sequences of bi-orthogonal vectors as the Lanczos process. The coefficients of the recurrence relations are obtained by factorizing a tridiagonal matrix without pivoting. This LU-factorization is an important source of instability which partially explains the erratic behaviour of the convergence curves of BCG. Moreover, it increases the danger of breakdowns, since the BCG iteration cannot be continued when the LU-decomposition does not exist.[21] To the best of our knowledge, no convergence formula is available for BCG.

The QMR iteration calls the (standard or look-ahead) Lanczos process to generate pairs of bi-orthonormal vectors. The recurrence relations are updated by applying a least-squares procedure 'quasi-minimization' to only one part of the residual expression. QMR can be viewed as a more stable implementation of BCG. The least squares procedure always has unique solution in contrast to the LU-factorization of BCG. Thus, an important source of breakdown and instabilities is eliminated. Freund and Nachtigal[10] proposed a convergence formula for QMR formally similar to the one given in equation (4). In contrast to GMRES, however, no convergence estimates based on this formula are employed in this work, since we could not evaluate it for the parameters of the model problems presented in the next section. The implemented version of the QMR algorithm is based on the two-term recurrencies approach without look-ahead developed by Freund and Nachtigal.[11]

Since GMRES minimizes the residual norm (equation (4)) in each step while BCG and QMR do not, we expect the former scheme to converge in a smaller number of iterations than the latter ones. BCG and QMR perform matrix–vector multiplications with A^T. When A has irregular sparsity, additional work and storage are needed to make A^T available for multiplication. This drawback of BCG and QMR motivated the development of Lanczos schemes that do not involve the transpose matrix.

2.3. Transpose-free Lanczos-type schemes (CGS, Bi-CGSTAB, TFQMR)

We have implemented three Lanczos-type schemes that do not involve matrix–vector products with A^T. While CGS is based upon a modification of the BCG scheme, Bi-CGSTAB and TFQMR can be viewed as further modifications of CGS. In exact arithmetic, all of them will break down every time BCG breaks down.

Sonnenveld[8] was the first to design a transpose free scheme, called the 'Conjugate Gradient Squared' (CGS) method. He observed that the scalar products of BCG can be rewritten in such a way that the matrix–vector multiplications with A^T are eliminated from the original BCG scheme. The residual vector $\mathbf{r}_k$ of the modified CGS scheme is proportional to the squared residual polynomial $p_k(A)$ of BCG. Since the rate of convergence of BCG depends upon $p_k(A)$ (conform equation (3)) and that of CGS upon $(p_k(A))^2$, we expect CGS to converge or diverge faster than BCG by a factor between one and two. Moreover, the convergence curves of CGS are even more erratic than those of BCG.

To smoothen the erratic behaviour of CGS, van der Vorst[9] replaced $(p_k(A))^2$ by a product of two polynomials, i.e. $p_k(A)\,s_k(A)$, where $s_k \in \prod_k$ is a polynomial of degree k, whose coefficients are updated recursively using a local steepest descent procedure. The convergence curves of the resulting scheme, called Bi-CGSTAB, are smoother indeed than those of CGS. However, Bi-CGSTAB is not necessarily more stable. The stagnation of the steepest descent component represents an additional source of breakdowns. Examples where CGS converges while Bi-CGSTAB does not are presented by Peters.[22] Note that examples demonstrating the opposite can be constructed as well (see, e.g. van der Vorst[9]).

Freund[12] proposed another smoothly converging variant of CGS called the Transpose Free Quasi-Minimal Residual (TFQMR) algorithm. The TFQMR scheme is based on the observation that CGS does not use all available search directions when it generates a new approximate solution. A quasi-minimal residual procedure, similar to that used by the original QMR method, determines the coefficients of the relation between the new iterate and the search directions. TFQMR can be obtained from CGS by changing a few lines of the code.

The convergence properties of Bi-CGSTAB and TFQMR are not well understood. Relying on the resemblance of these algorithms to CGS, we expect them to converge or diverge faster than BCG or QMR by a factor between one and two.

2.4. CG applied to normal equations (*CGNR*)

CGNR is nothing but the classic CG scheme of Herstenes and Stiefel applied to the system of equations

$$A^{\mathrm{T}}A\mathbf{c}=A^{\mathrm{T}}\mathbf{b}, \tag{6}$$

where A^{T} denotes the transpose of the coefficient matrix. While the letter 'N' in CGNR points at the fact that the coefficient matrix of equation (6) is normal, the letter 'R' suggests that this version manipulates the expression of the residual during the iteration. The code used in our tests is based on the 'adjusted' scheme of Bjoerk and Elfving.[23]

In contrast with the schemes described previously, CGNR seeks the kth approximate solution from the subspace $\mathbf{c}_0+K_k(A^{\mathrm{T}}\mathbf{r}_0, A^{\mathrm{T}}A)$. The 2-norm of the residual vector is minimized in each step

$$\|\mathbf{r}_k\|_2 = \min_{p\in\prod_k,\ p(0)=1} \|p(AA^{\mathrm{T}})\mathbf{r}_0\|_2. \tag{7}$$

Moreover, thanks to the CG connection, a three-term recurrence can be used to determine each new approximate solution.

Recall that the eigenvalues of the normal matrix AA^{T} are contained in the interval $\Sigma^2=[\sigma^2_{\min}, \sigma^2_{\max}]$, where $\sigma_{\min}$ and $\sigma_{\max}$ are the extreme singular values of A.

Replacing the polynomial p by the suitably scaled kth Chebyshev polynomial on Σ^2 and using some elementary estimates the following convergence formula is obtained:

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k, \tag{8}$$

where $\kappa(A)\equiv\sigma_{\max}/\sigma_{\min}$ is the condition number of A. Inequality equation (8) indicates that the convergence behaviour of CGNR is determined by the singular values of A rather than eigenvalues. Moreover, since the eigenvalues of AA^{T} are equal to the squared singular values of A, the convergence can be very slow for matrices with moderate to large condition numbers. Nevertheless, CGNR can be an effective choice for matrices with certain symmetries of the spectrum, like, e.g. shifted skew-symmetric matrices.[15]

3. TEST LINEAR SYSTEMS

3.1. Discretization techniques

We apply the Galerkin weighted residual method, as described by Pinder and Gray[24] to the advection–dispersion equation (1). The following matrix equation is obtained:

$$M\frac{\partial\mathbf{c}}{\partial t}+(V+D+Q+P)\mathbf{c}=Q\mathbf{c}_R, \tag{9}$$

where $\mathbf{c}$ contains the unknown values $c_n(t)$ of the concentration at the N nodes of the finite element mesh. The coefficients of $\mathbf{c}_R$ are known concentration values of the fluid entering the domain through some parts of the boundary ∂G. The entries of the matrices M, V, D, Q and P are given by

$$M_{mn}=\int_G \phi_m\phi_n \, \mathrm{d}G, \tag{10}$$

$$V_{mn}=\int_G v_i\phi_m\frac{\partial\phi_n}{\partial x_i}\, \mathrm{d}G, \tag{11}$$

$$D_{mn}=\int_G d_{ij}\frac{\partial\phi_m}{\partial x_j}\frac{\partial\phi_n}{\partial x_i}\, \mathrm{d}G, \tag{12}$$

$$Q_{mn}=\delta_{mn}\max\left[\int_{\partial G} n_i v_i\phi_n \, \mathrm{d}\partial G; 0\right], \tag{13}$$

$$P_{mn}=-\int_{\partial G} d_{ij}\frac{\partial\phi_m}{\partial x_j}\phi_n n_i \, \mathrm{d}\partial G, \tag{14}$$

where ϕ_n are Chapeau basis functions. The fluid velocity v_i and the dispersion coefficient d_{ij} in the expressions (10)–(14) are assumed to be piecewise constant over the elements. While n_i in equation (14) are the components of the outward pointing normal on the boundary ∂G, δ_{ij} in equation (13) denotes the Kronecker delta function.

Equation (9) is integrated in time using an implicit weighted finite difference scheme. A non-symmetric system of linear equations of the form given in equation (2) is obtained upon setting

$$A\equiv\left[\frac{1}{\Delta t}M+\theta(V+D+Q+P)\right],$$

$$\mathbf{b}\equiv\left[\frac{1}{\Delta t}M-(1-\theta)(V+D+Q+P)\right]\mathbf{c}^t+\theta Q\mathbf{c}_R^{t+\Delta t}+(1-\theta)Q\mathbf{c}_R^t, \tag{15}$$

and $\mathbf{c}\equiv\mathbf{c}^{t+\Delta t}$. The top notations $t+\Delta t$ and t denote the new and old time levels, Δt is the time step and θ is a weighting factor $(0{\cdot}5\leq\theta\leq 1{\cdot}0)$. Recall that the popular Crank–Nicolson scheme is obtained for $\theta=0{\cdot}5$.

It is well known that the 'consistent' FE formulation described by equations (9)–(15) is unconditionally stable but not always accurate. In particular when the advective components of equation (1) dominate and the discretization is not suitably chosen, oscillations of the solution occur. Daus *et al.*[25] showed that violations of the discretization criteria

$$C\leq\frac{P}{2}\quad\text{and}\quad C\leq 1 \tag{16}$$

lead to a gradual deterioration of the solution accuracy of the one-dimensional form of equation (1). While $P\equiv v\Delta x/d$ denotes the grid Peclet number, $C\equiv v\Delta t/\Delta x$ is the Courant number and Δx is the grid spacing. Their experiments suggest that the consistent FE formulation is more sensitive to violations of the latter criterion. In addition, Dause *et al.*[25] showed that discretizations for which the product

$$P_v=CP \tag{17}$$

is constant, produce the same error. $P_v\equiv v^2\Delta t/d$ is called the 'advective' Peclet number.

3.2. *Model problems*

The discretization techniques outlined in equations (9)–(15) are applied to the solution of three model problems. The *advancing front* problem and the diffusion in a plane *shear flow* problem are part of the set proposed as standard reference for accuracy measures by Baptista *et al.*[26] at the Convection–Diffusion Forum of the VII International Conference on Computational Methods in Water Resources. The convergence results obtained for the *advancing front* problem are also valid for the concentration hill problem of the same set, since the discretization of both problems yield the same coefficient matrix. The two-dimensional *plane dispersion* problem was employed by Kinzelbach and Frind[27] to illustrate the effects of grid anisotropy on the accuracy of finite element discretizations.

Since the aim of this work is to compare the capabilities of CG-like methods rather than perform accuracy tests, the specification data for the model problems were varied over a large range of values. Many of the employed discretization parameters produce accurate results, but some of them do not. The latter are used to demonstrate how the behaviour of the CG-like methods changes when the discretization is not suitably chosen.

Advancing front. This problem concerns the one-dimensional form of the advection–diffusion equation

$$\frac{\partial c}{\partial t}+v\frac{\partial c}{\partial x}-d\frac{\partial^2 c}{\partial x^2}=0, \tag{18}$$

with initial and boundary conditions

$$\begin{aligned} &c(x,t)=0 \quad t=t_0, \quad 0\leq x\leq L,\\ &c(x,t)=1 \quad t>t_0, \quad x=0,\\ &c(x,t)=0 \quad t>t_0, \quad x=L. \end{aligned}$$

The problem domain of length $L=1000$ is discretized in 100 elements of equal size $\Delta x=10$. While the magnitude of the velocity v is equal to unity, the time step Δt and diffusion parameter d are varied such that $1 \leq P \leq 10$ and $0{\cdot}2 \leq P_v \leq 10$.

The discussion in Section 4 is simplified without loss of generality by imposing a Dirichlet boundary condition in $x=L$. Any other type of boundary condition would rank-one modify the coefficient matrix. Eiermann *et al.*[3] showed that the consequence of this modification for the spectrum of eigenvalues of A is negligible for large values of N.

Shear flow. This problem concerns the special two-dimensional case of equation (1)

$$\frac{\partial c}{\partial t}+(v_0+vy)\frac{\partial c}{\partial x}-d\left(\frac{\partial^2 c}{\partial x^2}+\frac{\partial^2 c}{\partial y^2}\right)=0 \tag{19}$$

defined over a square domain with the side length $L=4000$ and the initial and boundary conditions

$$\begin{aligned} &c(x,y,t)=0 \quad t=t_0, \quad 0\leq x\leq L,\ 0\leq y\leq L,\\ &c(x,y,t)=0 \quad t>t_0, \quad x=L/5,\ y=L/2. \end{aligned}$$

The domain is discretized in triangles obtained by halving rectangular elements of side length $\Delta x=\Delta y=200$. The parameters defining the velocity field $v_0=0$ and $v=5\times 10^{-4}$ remain unchanged for all experiments. The time step and diffusion parameter are defined such that the

Peclet number and advective Peclet number vary within the intervals

$$P \equiv \bar{v}_x \Delta x/d \in [1, \ldots, 10] \quad \text{and} \quad P_v \equiv \bar{v}_x^2 \Delta t/d \in [0{\cdot}2, \ldots, 10],$$

where $\bar{v}_x \equiv v_0 + vL$ represents the maximal velocity.

Plane dispersion. This problem concerns the two-dimensional form of the advection–dispersion equation (equation (1)) defined over a square of length L. Scheidegger's formulas[1] are employed to calculate the dispersion coefficients. Thus, for the components of the velocity field $v_x = v_y = 1$ and the longitudinal and transversal dispersivity coefficients equal to 100 and 10, the entries of the dispersion matrix are $d_{xx} = d_{yy} = 110/\sqrt{2}$ and $d_{xy} = 90/\sqrt{2}$, respectively. The initial and boundary conditions and the intervals of variation for $P \equiv v_x \Delta x/d_{xx}$ and $P_v \equiv v_{xx}^2 \Delta t/d_{xx}$ are identical to those of the *shear flow* problem. Again, right isosceles of equal length are used for discretization, but this time the side length $\Delta x = \Delta y = L/20$ and time step are varied.

4. CONVERGENCE ESTIMATES

We apply GMRES and CGNR to the solution of the *advancing-front* problem and estimate the convergence rates as a function of the discretization parameters.

4.1. GMRES

The discretization of the advancing front problem yields the tridiagonal Toeplitz matrix

$$A = \text{tridiag}(\gamma, \alpha, \beta) \in \mathbb{R}^{N \times N}, \tag{20}$$

where for $\theta = 0{\cdot}5$ the non-zero entries of A are

$$\alpha = \frac{4P + 6C}{3C},$$

$$\beta = \frac{(3C+2)P - 6C}{6C},$$

$$\gamma = \frac{(-3C+2)P - 6C}{6C}.$$

The eigenvalues of A are given by

$$\lambda_j = \alpha + 2\sqrt{(\beta\gamma)} \cos\left(\frac{\pi j}{N+1}\right) \quad (j = 1, 2, \ldots, N), \tag{21}$$

where we neglected a term of order $(1/N)^2$.

We begin the calculation of the upper-bound of convergence of GMRES given in equation (5) with the approximation of the quantity $\min_{p \in \prod_k, p(0)=1} \| p(A) \|_2$. The two cases are distinguished as follows:

(a) The eigenvalues of A are real if and only if $\beta\gamma \geq 0$, or equivalently, if $2|P - 3C| \geq 3PC$ is satisfied. In this case, it follows that

$$\lambda_{\min}(A) \simeq \alpha - 2\sqrt{(\beta\gamma)} \quad \text{and} \quad \lambda_{\max}(A) \simeq \alpha + 2\sqrt{(\beta\gamma)}.$$

(b) For $\beta\gamma < 0$ (or equivalently, $2|P - 3C| < 3PC$) the eigenvalues of A are no longer real. They are contained in an interval perpendicular to the real axis, namely in $[\alpha - i\mu, \alpha + i\mu]$, where $i^2 = -1$ and

$$\mu \simeq 2\sqrt{(-\beta\gamma)}.$$

Relying on the assumption that the eigenvalues λ_j of A are smoothly distributed over $I\equiv[\lambda_{\min}(A), \lambda_{\max}(A)]$ if $\beta\gamma>0$, or over $J\equiv[\alpha-\mathrm{i}\mu, \alpha+\mathrm{i}\mu]$ if $\beta\gamma<0$, we replace $\max_{1\leq j\leq N}|p(\lambda_j)|$ by either $\max_{z\in I}|p(z)|$ if $\beta\gamma>0$, or $\max_{z\in J}|p(z)|$ if $\beta\gamma<0$. Moreover, using the results provided by Eiermann *et al.*,[28] we obtain

$$\overline{\lim_{k\to\infty}}\left(\min_{p\in\prod_k,\, p(0)=1}[\max_{z\in I}|p(z)|]\right)^{1/k}=\chi_1\equiv\frac{2\sqrt{(\beta\gamma)}}{\alpha+\sqrt{(\alpha^2-4\beta\gamma)}}$$
$$\overline{\lim_{k\to\infty}}\left(\min_{p\in\prod_k,\, p(0)=1}[\max_{z\in J}|p(z)|]\right)^{1/k}=\chi_2\equiv\frac{\sqrt{(\alpha^2-4\beta\gamma)}-\alpha}{2\sqrt{(-\beta\gamma)}}. \qquad (22)$$

To calculate the quantity $\kappa(T)$ in expression (5) we proceed as follows:

(a) First, observe that the eigenvectors of A are identical to those of the matrix tridiag$(\gamma, 0, \beta)$. Using the results presented by Eiermann[29] (Ch. 1) we write that the eigenvector corresponding to λ_j has the expression

$$t_j=\left(\delta^1\sin\left(\frac{\pi j}{N+1}\right),\ldots,\delta^N\sin\left(\frac{N\pi j}{N+1}\right)\right)^{\mathrm{T}}\in\mathbb{C}^N, \qquad (23)$$

where $\delta=\sqrt{(|\gamma/\beta|)}$.

(b) Then, write the matrix of eigenvectors $T=[\mathbf{t}_1,\ldots,\mathbf{t}_N]$ as the product of the matrices $D=\mathrm{diag}(\delta^1,\ldots,\delta^N)$ and $V=[\mathbf{v}_1,\ldots,\mathbf{v}_N]$, where

$$\mathbf{v}_j=\left(\sin\left(\frac{\pi j}{N+1}\right),\ldots,\sin\left(\frac{N\pi j}{N+1}\right)\right)^{\mathrm{T}}\in\mathbb{C}^N.$$

(c) Finally, using simple manipulations based on the identity

$$V^{-1}=\frac{2}{N+1}V^{\mathrm{T}}=\frac{2}{N+1}V,$$

we arrive at the following expression for the condition number of T

$$\kappa(T)\equiv\|T\|_2\,\|T^{-1}\|_2=\tau^{N-1}, \qquad (24)$$

where $\tau=\max\{\delta, 1/\delta\}$.

The substitution of $\kappa(T)$ (equation (24)) and $\min_{p\in\prod_k,\, p(0)=1}\|p(\tilde{A})\|_2$ (equation (22)) in equation (5) gives the following expression

$$\bar{k}=\frac{(1-N)\log_{10}\tau-\eta}{\log_{10}\chi}, \qquad (25)$$

where $\bar{k}$ is the ceiling on the iteration index k required to reduce the residual norm by the factor $10^{-\eta}$. χ is equal to χ_1 for $\sqrt{(\beta\gamma)}>0$ and to χ_2 for $\sqrt{(\beta\gamma)}<0$ (equation (22)). (The non-relevant special case $\beta\gamma=0$, when matrix A is not diagonalizable, was omitted.)

Peters[30] showed that the convergence estimate given in equations (5) and (25) approximates the behaviour of GMRES accurately only when $\kappa(T)$ does not differ too much from unity, i.e. the matrix A is not too far from normal. This result confirms the elegant theory developed by Trefethen[18] which asserts that for arbitrary matrices, for which $\kappa(T)$ is huge, the scalar approximation problem expressed by equation (5) should be redefined in terms of pseudo-spectra. Unfortunately, the convergence estimate based on the pseudo-spectrum of A is still too inaccurate with respect to our problem. In contrast, we found that the empirical formula

$$\bar{k}_1 = \frac{-\eta}{\log_{10}\Lambda + \log_{10}\tau} \tag{26}$$

gives strikingly accurate estimates. We obtained equation (26) by (. . . mistakingly) using k as exponent instead of $N-1$ in equation (24). Figure 1(a) shows the relation between $\bar{k}_1$ and the Peclet number and advective Peclet number for $\eta = 10$ (see Figure 2(a) for comparison).

4.2. *CGNR*

The upper bound of convergence of CGNR given by equation (8) is a function of the condition number of A. To compute $\kappa(A)$, we build explicitly the normal matrix

$$A^{\mathrm{T}}A \equiv \mathrm{pentadiag}(\mathrm{c}, \mathrm{b}, \mathrm{a}, \mathrm{b}, \mathrm{c}), \tag{27}$$

where A is the tridiagonal Toplitz matrix given in equation (20) and the entries of the normal matrix in equation (27) are functions of the Peclet number and Courant number

$$\mathrm{a} = \frac{(C^2+4)P^2 + 8CP + 12C^2}{2C^2},$$

$$\mathrm{b} = \frac{8P^2 - 12CP - 36C^2}{2C^2},$$

$$\mathrm{c} = \frac{(-9C^2+4)P^2 - 24CP + 36C^2}{36C^2}.$$

The eigenvalues of the normal matrix AA^{T} are given by the expression

$$\lambda_j = \mathrm{a} + 2\mathrm{b}\cos\left(\frac{\pi j}{N+1}\right) + 2\mathrm{c}\cos\left(\frac{2\pi j}{N+1}\right), \tag{28}$$

where, a term of order $(1/N)^2$ is neglected (see, e.g. Reference 31).

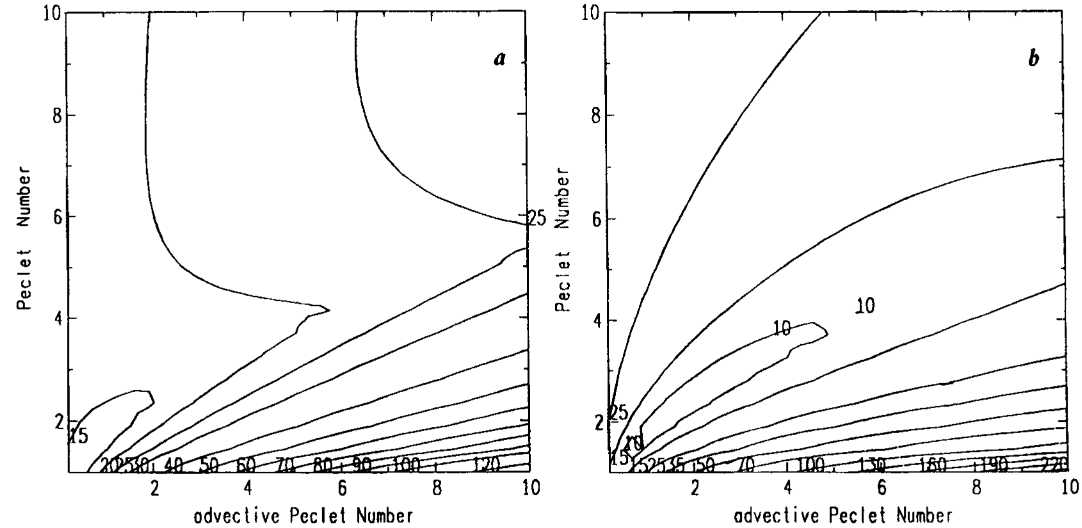

Figure 1. (a) GMRES; (b) CGNR. The dependency of the convergence estimates $\bar{k}_1$ (equation (26)) and $\bar{k}_2$ (equation (30)) upon the Peclet number and advective Peclet number, for $\eta = 10$

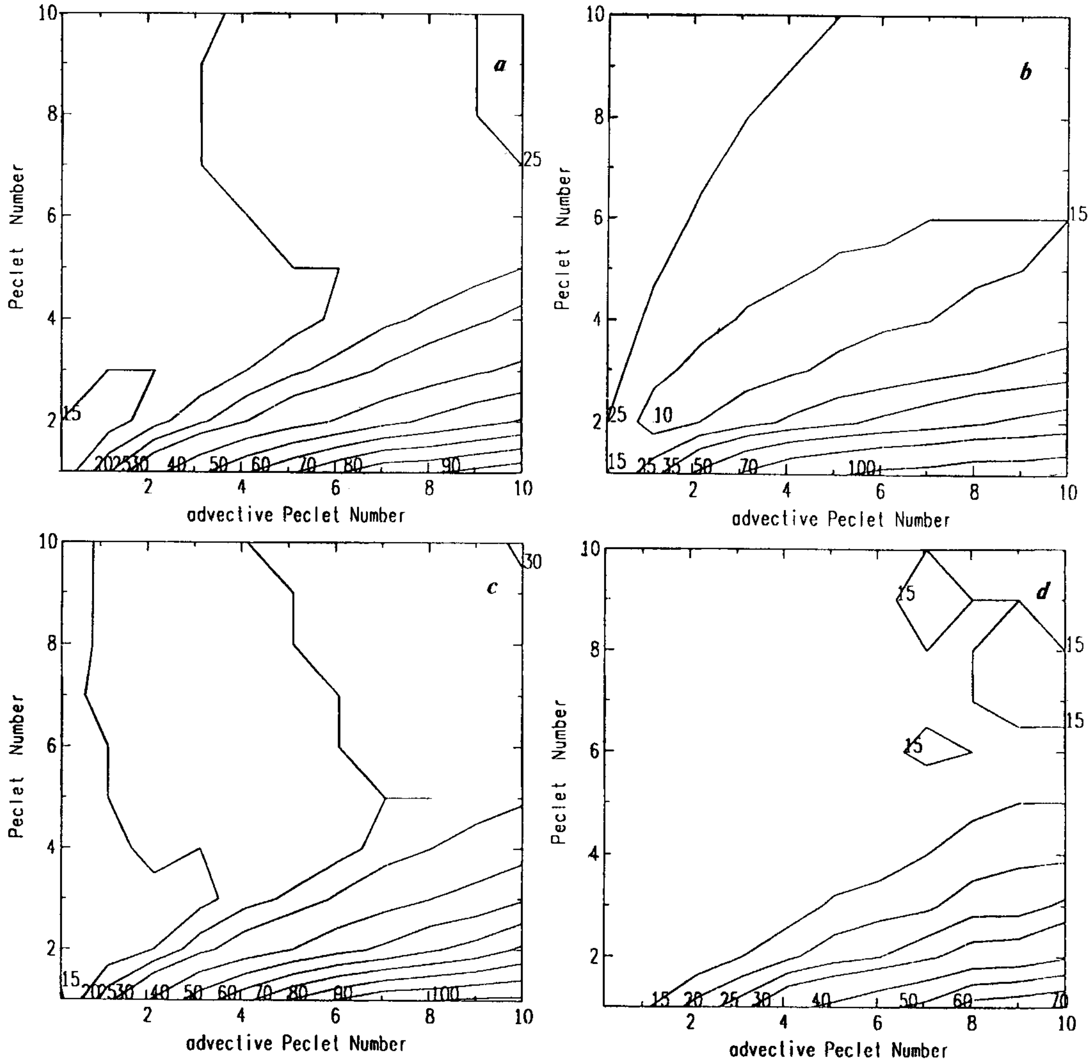

Figure 2. *Advancing front*: (a) GMRES; (b) CGNR; (c) BCG; (d) CGS. The dependency of the number of iterations k upon the Peclet number and advective Peclet number, for $\|r_k\|_2 \leq 10^{-10}$

The condition number of A is given by one of the following expressions:

$$\kappa(A)=\sqrt{\left[\frac{\max(a+2c\pm 2b, a-2c-b^2/4c)}{\min(a+2c\pm 2b, a-2c-b^2/4c)}\right]} \quad \text{for } |b|\leq|4c|$$

$$\kappa(A)=\sqrt{\left[\frac{\max(a+2c\pm 2b)}{\min(a+2c\pm 2b)}\right]} \quad \text{for } |b|>|4c|. \tag{29}$$

To obtain the ceiling of the iteration index k required to reduce the residual norm by a factor

$10^{-\eta}$, we introduce the value of $\kappa(A)$ computed in equation (29) into the expression

$$\bar{k}_2 = \frac{-\eta}{\log_{10}\left(\dfrac{\kappa(A)-1}{\kappa(A)+1}\right)}. \tag{30}$$

Figure 1(b) shows the surface plot of $\bar{k}_2$ as a function of the Peclet number and advective Peclet number, for $\eta = 10$. The corresponding experimental results are shown in Figure 2(b).

4.3. Reducing the number of iterations

A major question for any practitioner who applies GMRES or CGNR to the solution of equation (9) is how to select the discretization parameters such that the solution of desired accuracy is obtained in the smallest number of iterations. Figure 1 indicates that the ceiling functions of GMRES $\bar{k}_1$ (equation (26)) and CGNR $\bar{k}_2$ (equation (30)) build up local minima along the parabola

$$P_v = P^2/3. \tag{31}$$

Equation (31) holds when $\kappa(T)$ defined in equation (24) is equal to unity, i.e. the coefficient matrix A is normal. Note that discretization parameters satisfying this 'normality'condition, or equivalently $P = 3C$, always fulfill the first accuracy criterion given in equation (16).

Recall that the accuracy of the Crank–Nicolson scheme increases for small time steps (see, e.g. Reference 32). Moreover, Figure 1 suggests that the selection of small time steps accelerate the convergence of GMRES and CGNR. This strategy is not necessarily advantageous since it increases the number of time steps and linear systems accordingly. The discretization parameters yielding a solution at the time $t =$ constant, in the smallest number of iterations can be estimated for GMRES and CGNR by claculating the minima of the functions

$$\hat{k}_1 = \bar{k}_1/\Delta t \tag{32}$$

and

$$\hat{k}_2 = \bar{k}_2/\Delta t, \tag{33}$$

respectively. Similar to the functions plotted in Figure 1, $\hat{k}_1$ and $\hat{k}_2$ have local minima along the parabola (31).[30]

Up to this point, we focused on the convergence properties of GMRES and CGNR. Despite the simplicity of the problem we were not able to obtain an analytic estimate for the convergence of QMR. The fact that BCG and QMR converge in a larger number of iterations than GMRES and the assumption that CGS, Bi-CGSTAB and TFQMR converge about twice as fast as BCG are too vague estimates for reliable predictions. Nevertheless, since the Lanczos-type schemes and GMRES construct iterates in the same Krylov subspace, we expect them to respond similarly to changes of the discretization parameters.

5. NUMERICAL EXPERIMENTS

5.1. Model problems

We present the results of several convergence tests performed with systems of equations stemming from the discretization of the problems described in Section 3. The surface plots in Figures 2–4 are obtained by interpolating the iteration number k required to reduce the residual norm to $\|\mathbf{r}_k\|_2 \leq 10^{-10}$. Each problem employs the same random starting vector. The right-hand side vector and starting vector are scaled such that $\|\mathbf{r}_0\|_2 = 1$.

Advancing front. Figure 2 illustrates the relation between the iteration indices of GMRES, CGNR, BCG and CGS and the parameters P and P_v. No preconditioning is employed, since the goal of this exercise is to validate the convergence formulas (26) and (30). As expected, the form of the interpolated surfaces of BCG and CGS is similar to that obtained for GMRES. While GMRES executes slightly less iterations than BCG, CGS converges about twice as fast.

Shear flow and plane dispersion. The matrices stemming from the discretization of these two problems, are diagonally scaled. Instead of solving the linear equations (2), we apply the CG-like algorithms to the system

$$\tilde{M}^{-1}A\mathbf{c} = \tilde{M}^{-1}\mathbf{b}, \tag{34}$$

where $\tilde{M} = \text{dia}\sqrt{(A^{\mathrm{T}}A)}$.

To obtain the same order of complexity for all implemented schemes, i.e. number of floating point operations per iteration, the restarted version GMRES(20) is employed. Table I lists the relative amounts of time per iteration for all schemes. The values are obtained by dividing the measured CPU-time by the number of iterations for each scheme and then scaling the results with respect to the smallest one obtained for CGNR. The coefficient matrix of order 421 with 9 non-zero diagonals is stored in 'compressed diagonal' form.[33] Since the time required to perform matrix–vector multiplications with A and A^{T} is the same for this particular storage scheme, the values displayed for BCG, QMR and CGNR in Table I look rather optimistic.

Figure 3 shows the dependency of the number of iterations of GMRES(20), CGNR, BCG and CGS upon the Peclet number and advective Peclet number for the *shear flow* problem. Figure 4 illustrates the variation of the iteration indices of GMRES(20), CGNR, BCG and CGS for the *plane dispersion* problem. The convergence histories of QMR, Bi-CGSTAB and TFQMR are very similar to those of BCG and CGS and were not shown. Samples of the convergence histories of all implemented methods for the *plane dispersion* problem have been illustrated by Peters.[30]

The experimental results do not differ from the analytic ones presented in the previous section. GMRES(20) requires more iterations than BCG and QMR when the iteration index is larger than the restart one since the global minimization property (equation (4)) is lost. CGNR is more exposed to over-unitary values of the Courant number than GMRES and the Lanczos schemes. The transpose-free Lanczos schemes converge in the smallest number of iterations.

Table I. Relative time/iteration

Scheme	Value
GMRES(20)	1·18
BCG	1·06
QMR	1·20
CGS	1·03
Bi-CGSTAB	1·09
TFQMR	1·13
CGNR	1·00

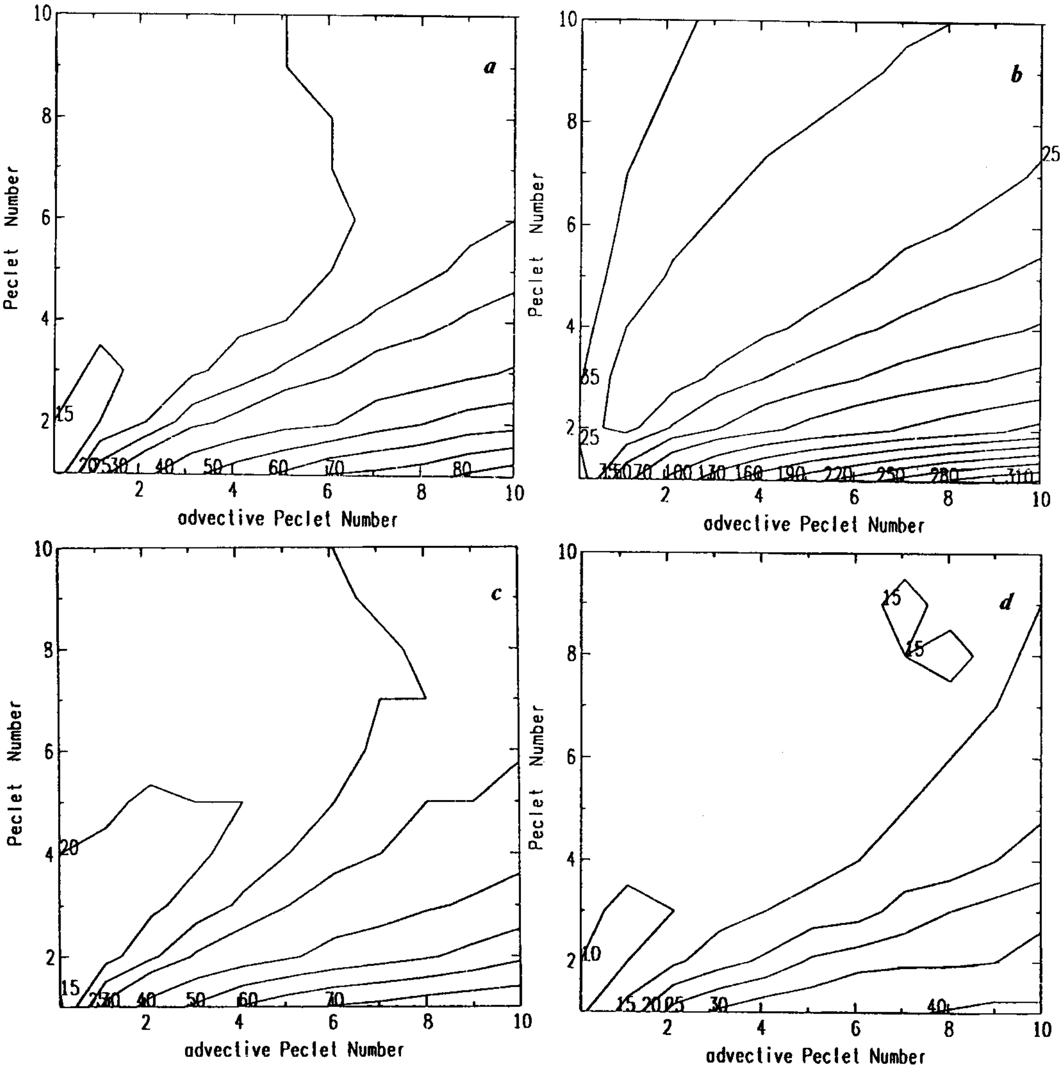

Figure 3. *Shear flow*: (a) GMRES; (b) CGNR; (c) BCG; (d) CGS. The dependency of the number of iterations k upon the Peclet number and advective Peclet number, for $\|r_k\|_2 \leq 10^{-10}$

5.2. *Particular cases*

Three particular cases based on the *plane dispersion* problem are considered:

The explicit case. The time step and the parameter θ act as weighting factors for the components of the coefficient matrix A (equation (15)). We observed that, if instead of the Crank–Nicolson scheme the fully implicit scheme ($\theta = 1$) is chosen, the convergence speed of the CG-like schemes decreases. In contrast, if the explicit scheme ($\theta = 0$) is selected, the convergence is optimal. The convergence history of all schemes is shown for the latter case in Figure 5(a). Matrix A is identical to the SPD matrix M (defined in equation (10)), while BCG and QMR become equivalent to the classical CG method. The eigenvalues of A are located on the real axis and far

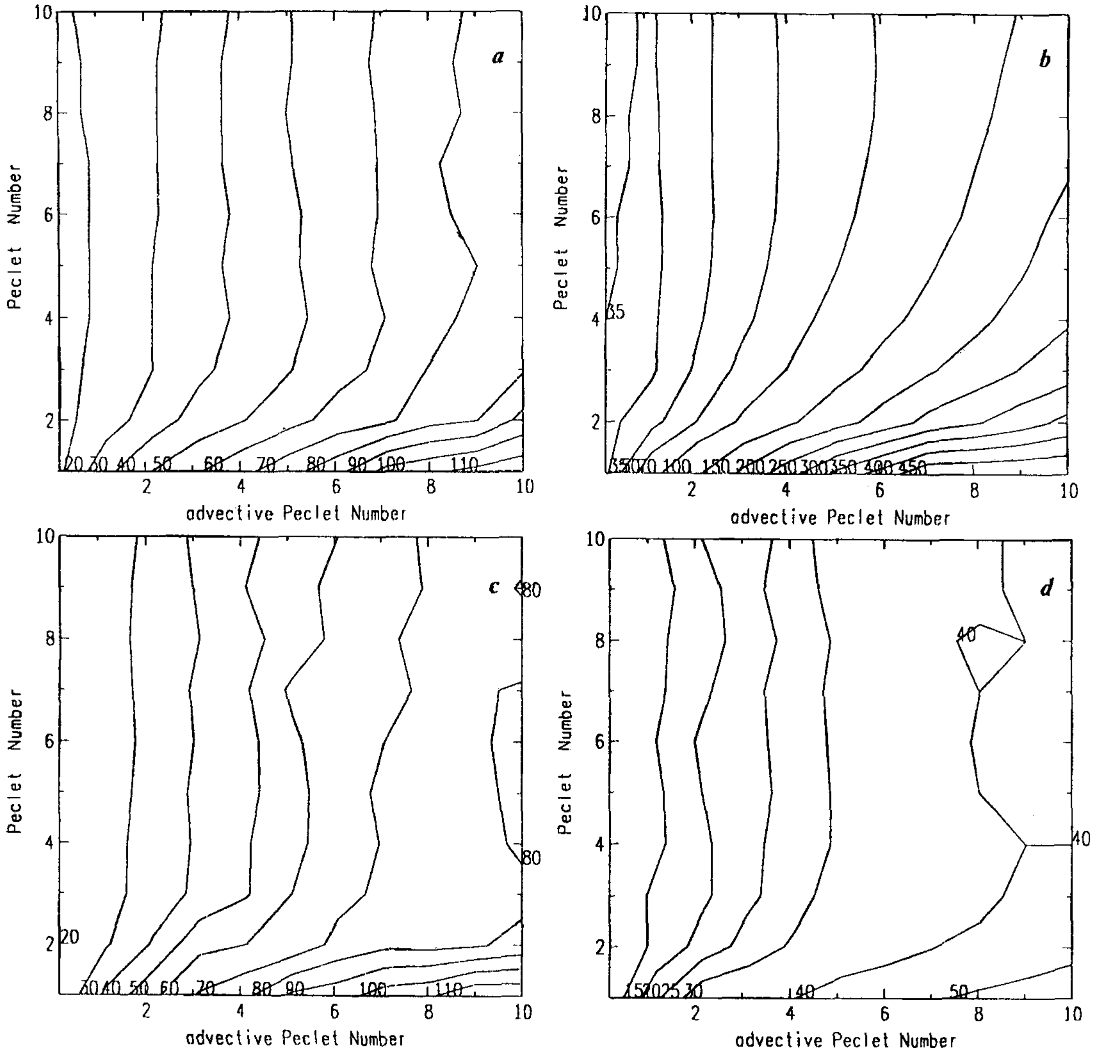

Figure 4. *Plane dispersion*: (a) GMRES; (b) CGNR; (c) BCG; (d) CGS. The dependency of the number of iterations k upon the Peclet number and advective Peclet number, for $\|r_k\|_2 \leq 10^{-10}$

enough from origin (Figure 5(b)). Recall that the conditional stability of the explicit scheme requires the selection of small time steps. When the number of time steps is large, the total number of iterations may not be optimal any more.

The pure dispersion case. Small Peclet numbers emphasize the presence of the dispersion component D in the coefficient matrix A. Matrix D has real positive eigenvalues but this is not necessarily an asset for convergence. Figure 6(a) presents the convergence histories for the worst case involving small Peclet and large Courant numbers: the Courant number is infinite, the Peclet number is equal to zero and equation (1) reduces to the steady-state diffusion equation. The spectrum of A, illustrated in Figure 6(b), is uniformly distributed within a segment located on the positive half of the real axis, close to the origin. While the Lanczos methods require more than

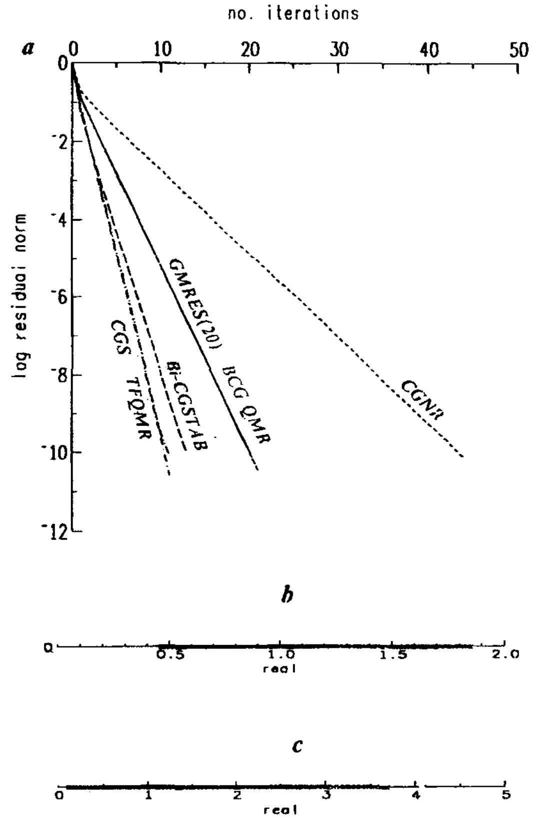

Figure 5. The explicit case: $\theta=0$. (a) convergence histories; (b) the eigenvalues of A; (c) eigenvalues of AA^{T}

one hundred iterations (Bi-CGSTAB is the fastest), GMRES(20) converges monotonically but very slowly. CGNR stagnates almost for more than 500 iterations before converging. In both cases presented in Figures 5 and 6, the eigenvalues of $A^{\mathrm{T}}A$ squares those of A.

The pure advection case. This is the worst case for Lanczos schemes. The convergence histories are presented in Figure 7(a). An almost skew-symmetric matrix is obtained by making coefficient matrix A identical with the advection matrix V. The Peclet and Courant numbers are equal to infinity. The eigenvalues of A are located around the imaginary axis, as shown in Figure 7(b). Neither GMRES(20) nor the Lanczos methods converge. The only converging scheme is CGNR. The eigenvalues of matrix $A^{\mathrm{T}}A$ are shown in Figure 7(c).

6. FINAL REMARKS

The purpose of this work was to understand the behaviour of non-symmetric CG-like schemes in the context of solving the advection–dispersion equation by the finite element method. We developed convergence estimates for some methods, applied them to selected reference problems and presented empirical evidence in support of the estimates. In addition, examples were constructed to illustrate the difference in capabilities between schemes. In order to obtain

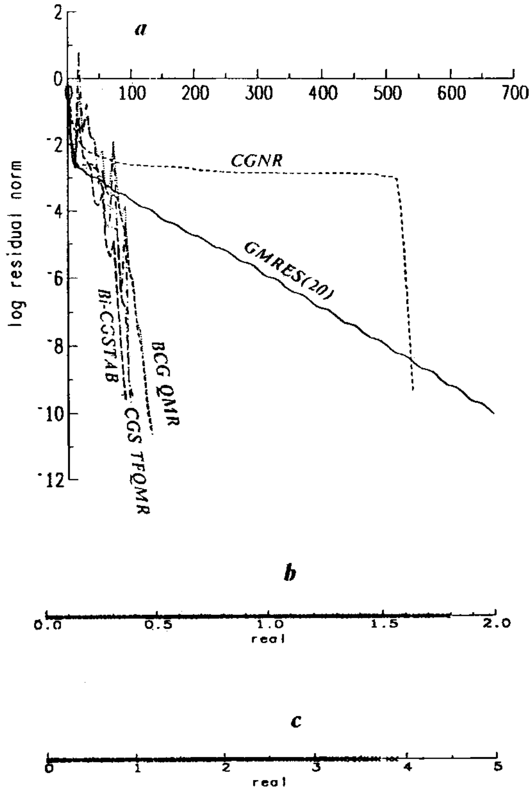

Figure 6. The pure dispersion case: $P=0$, $C\to\infty$. (a) convergence histories; (b) the eigenvalues of A; (c) eigenvalues of AA^T

a non-distorted picture of the relation between discretization and convergence, we employed neither 'hard' preconditioners, like incomplete factorizations, nor procedures that avoid breakdowns, like look-ahead.

All considered CG-like methods converge properly when the time and space discretization yield the accurate solution of the advection–dispersion equation. The fulfillment of the accuracy criterion (16) and normality condition (31) leads to an almost optimal convergence behaviour of all algorithms. If only few iterations are performed and the storage requirement does not increase too much, GMRES is a good choice. The non-restarted version minimizes the residual norm in each iteration step. Moreover, only one matrix–vector multiplication per iteration is needed. For problems involving more iterations, the transpose-free Lanczos schemes may be the best choice. They converge between one and two times faster than the other Lanczos schemes, perform a constant amount of work per iteration and can be implemented easily. Table I shows that CGS needs less time per iteration than other schemes. CGNR may be the only converging scheme for advective dominant problems, where the coefficient matrix is almost skew symmetric. However, CGNR squares the condition number of the coefficient matrix and performs poorly in most cases.

Unfortunately, the application of the criteria (16) and (31) is not always possible. In the case of more complex problems, an obvious way to improve the convergence of the CG-like schemes is to decrease the size of the time step. The time step weights the contributions from the SPD mass matrix M to the coefficients of A. The eigenvalues of matrix A are not determined solely by the

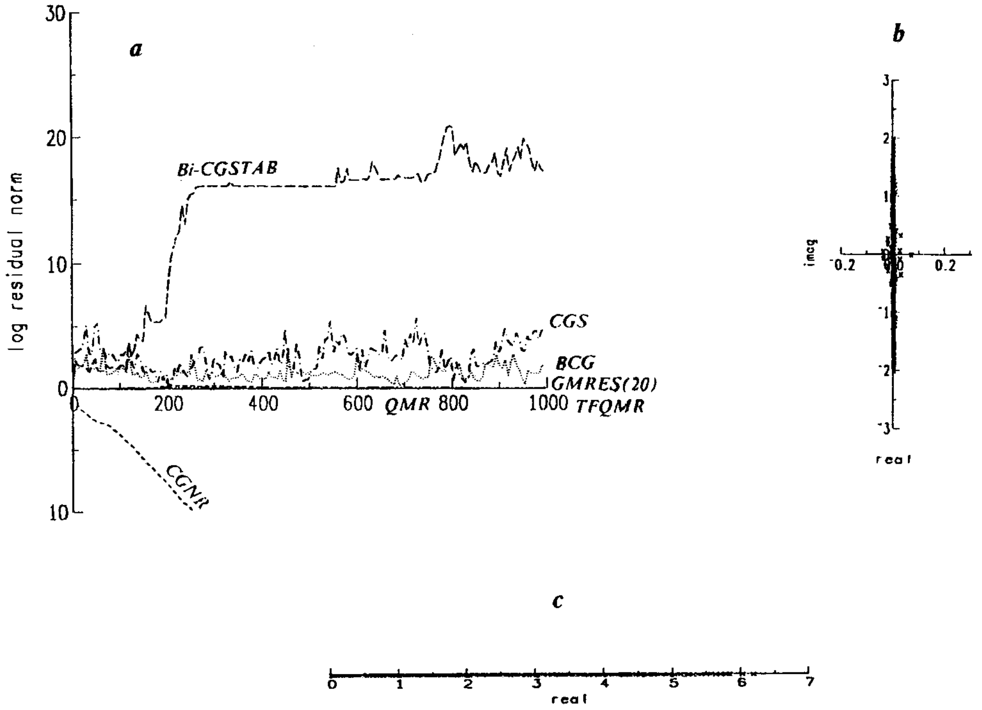

Figure 7. The pure advection case: $P \rightarrow \infty$, $C \rightarrow \infty$. (a) convergence histories; (b) the eigenvalues of A; (c) eigenvalues of AA^T

entries of M (equation (10)). However, large relative contributions from this SPD matrix keep the eigenvalues and singular values of A far from the origin of the co-ordinates system and ensure the fast convergence of CG-like methods. (Note that the use of the Crank–Nicolson scheme instead of the full implicit time marching scheme has a similar effect.)

To demonstrate the importance of the time weighting, we consider an additional example: The *shear flow* problem is defined on a square domain of side length $L=4000$. This time isosceles of side length ($\Delta x=\Delta y=20$) are used and a system of linear equations with 40401 unknowns is obtained. A random function $c_0(x, y, t_0)$ with values between zero and unity describes the initial condition. The parameters of the velocity field are $v_0=1$ and $v=1\times10^{-3}$ (see equation (19)). The time step is $\Delta t=20$ and the diffusion coefficient is equal to unity. The grid Peclet number varies along the vertical y-direction from 4 (for $y=0$) to 20 (for $y=4000$) while the Courant number goes from 1 to 5.

The upper part of Table II shows the CPU-time measurements and the number of iterations required by CGS and GMRES(20) to reduce the residual norm to 10^{-6}. The experiments are performed on an IBM 3090 VF. The diagonal scaling (D) (cf. equation (34)) and an incomplete factorization (ILU) of the matrix are used as preconditioners.[32] CGS converges slowly and only after the ILU preconditioner is employed. GMRES(20) works better, but it still requires a large number of iterations. The lower part of Table II shows convergence results obtained for the same problem by halving the time step ($\Delta t=10$). The smaller time step improves the accuracy and robustness of the solution and accelerates the convergence. The ILU preconditioner reduces further the number of iterations of both methods. Moreover, it helps GMRES to converge

Table II. Convergence results

Scheme	No. iterations	CPU time (s)
D-CGS	—	—
ILU-CGS	226	13·42
D-GMRES(20)	154	6·46
ILU-GMRES(20)	74	3·55
D-CGS	44	1·83
ILU-CGS	10	0·60
D-GMRES(20)	53	2·22
ILU-GMRES(20)	13	0·62

without restarts. Even with twice as many time steps, the latter tests run much faster than the former ones.

The experience of many researchers has proven that CG-like algorithms work effectively for different classes of linear systems. However, when applied to the finite element solution of the advection–dispersion equation, the efficiency of these schemes should not be taken for granted. If the discretization is not done with care, the grid Courant and Peclet numbers are high, CG-like algorithms perform poorly or even break down. The first and most natural step to improve the convergence is to change the discretization parameters such that a well-conditioned matrix is obtained. If then an efficient preconditioner is applied to the well-posed problem, the iterative schemes work very efficiently. Slow convergence should be a matter of concern not only because the solution requires much computer time, something may be wrong with the accuracy, as well.

AKNOWLEDGEMENT

The author is indebted to Helmut Daniels, Michael Eiermann, Noel Nachtigal, Richard Reuter and Ulrich Scharffenberger for helpful discussions.

REFERENCES

1. J. Bear, *Dynamics of Fluids in Porous Media*, Elsevier, New York, 1972.
2. M. R. Hestenes and E. Stiefel, 'Methods of conjugate gradients for solving linear systems', *J. Res. Nat. Bur. Standards*, **49**, 409–436 (1952).
3. M. Eiermann, A. Peters and H. Daniels, 'Computational complexity analysis of two Galerkin-FE/CG-like approaches for the advection–dispersion equation', *SIAM J. Sci. Stat. Comput.*, 1991 (submitted).
4. H. M. Leismann and E. O. Frind, 'A symmetric-matrix time integration scheme for the efficient solution of advection–dispersion problems', *Water Resources Res.*, **25**, 1133–1139 (1989).
5. C. Lanczos, 'Solution of systems of linear equations by minimized iterations', *J. Res. Natl. Bur. Standards*, **45**, 255–282 (1952).
6. R. Fletcher, 'Conjugate gradient methods for indefinite systems', in H. Watson (ed.), *Lecture Notes in Math.*, Springer, Berlin, 1976, **506**, pp. 73–89.
7. Y. Saad and M. H. Schultz, 'GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
8. P. Sonneveld, 'CGS, a fast Lanczos-type solver for nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **10**, 36–52 (1989).
9. H. A. van der Vorst, 'Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **13**, 631–644 (1992).
10. R. W. Freund and N. M. Nachtigal, 'QMR: a quasi-minimal residual method for non-Hermitian linear systems', *Numerische Math.*, **60**, 315–339 (1991).
11. R. W. Freund and N. M. Nachtigal, 'An implementation of the QMR method based on coupled two-term recurrences', *RIACS Technical Report 92.15*, NASA Ames Research Center, Moffett Field, 1992.

12. R. W. Freund, 'A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems', *RIACS Technical Report 91.18*, NASA Ames Research Center, Moffett Field, 1991.
13. Y. Saad, 'Krylov subspace methods on supercomputers', *SIAM J. Sci. Stat. Comput.*, **10**, 1200–1232 (1989).
14. H. A. van der Vorst, 'Iterative methods for the solution of large systems on supercomputers', *Adv. Water Resources*, **13**, 137–146 (1990).
15. R. W. Freund, G. H. Golub and N. M. Nachtigal, Iterative Solution of Linear Systems, *Acta Numerica*, 1–44 (1991).
16. N. M. Nachtigal, S. C. Reddy and L. N. Trefethen, 'How fast are nonsymmetric matrix iterations?, *SIAM J. of Matrix Analysis and Applications.* **13**(3), 778–795 (1992)
17. V. Faber and T. Manteuffel, 'Necessary and sufficient conditions for the existence of a conjugate gradient method,' *SIAM J. Numer. Anal.*, **21**, 352–362 (1984).
18. L. N. Trefethen, 'Approximation theory and numerical linear algebra', in J. C. Mason and M. G. Cox (eds), *Algorithms and Approximation II*, Chapman and Hall, London, 1990, pp. 336–360.
19. G. H. Golub and C. F. van Loan, *Matrix Computations*, 2nd edn, Johns Hopkins University Press, Baltimore, 1989.
20. R. W. Freund, M. H. Gutknecht and N. M. Nachtigal, 'An implementation of the look-ahead Lanczos algorithm for non-hermitian matrices, *RIACS Technical Report*, 90.45, NASA Ames Research Center, Moffett Field, 1990.
21. M. H. Gutknecht, 'The unsymmetric Lanczos algorithms and their relations to Pade approximations, continued functions and the QD algorithm', *Proc. Copper Mountain Conf. Iterative Methods*, University of Colorado, Denver, 1990.
22. A. Peters, 'CG-like algorithms for linear systems stemming from the FE discretization of the advection–diffusion equation', *Proc. Copper Mountain Conf. Iterative Methods*, University of Colorado, Denver, 1992.
23. A. Bjorck and T. Elfving, 'Accelerated projection methods for computing pseudo-inverse solutions of systems of linear equations', *BIT*, **19**, 145–163 (1979).
24. G. F. Pinder and W. G. Gray, *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic Press, New York, 1977.
25. A. D. Daus, E. O. Frind and E. A. Sudicky, 'Comparative error analysis in finite element formulations of the advection–dispersion equation', *Adv. Water Resources*, **8**, 86–95 (1985).
26. A. Baptista, P. Gresho and E. Adams, 'Convection diffusion forum', *Proc VIII Int. Conf. Numerical Methods in Water Resources*, MIT Press, Cambridge, 1988, personal communication.
27. W. K. H. Kinzelbach and E. O. Frind, 'Accuracy criteria for advection—dispersion models', in A. Sa da Costa *et al.* (eds.), *Proc. VI Internat. Conf. FE in Water Resources*, Springer, Berlin, 1986, pp. 489–501.
28. M. Eiermann, W. Niethammer and R. S. Varga, 'A study of semiiterative methods for nonsymmetric systems of linear equations', *Numer. Math.*, **47**, 505–533 (1985).
29. M. Eiermann, Semi-iterative Verfahren fuer nichtsymmetrische lineare Gleichungssysteme, *Habilitationsschrift*, Universitaet Karlsruhe, 1989.
30. A. Peters, 'Nonsymmetric CG-like schemes and the FE solution of the advection-dispersion equation', *IBM Heidelberg Technical Report 75.92.24*, 1992.
31. L. Berg, *Lineare Gleichungssysteme mit Bandstruktur*, Carl Hanser, Munich, 1986.
32. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*, Vol. 1, Springer, Berlin, 1988.
33. IBM ESSL, *Engineering and Scientific Subroutine Library-Guide and Reference Release* 4, SC23-0184-4, IBM Corporation, 1990.