MATHEMATICS DEPARTMENT

Numerical Experiences with Bi-CGSTAB on an Advection-Diffusion Problem

K.J. Neylon

Numerical Analysis Report 8/93

UNIVERSITY OF READING

Numerical Experiences with Bi-CGSTAB on an Advection-Diffusion Problem

K.J. Neylon

Numerical Analysis Report 8/93

Department of Mathematics
P.O. Box 220
University of Reading
Whiteknights
Reading
RG6 2AX
United Kingdom

Abstract

The Bi-CGSTAB nonsymmetric linear solver has the attractive conjugate gradient-like properties of efficiency, low storage and no external parameters. However, unlike the conjugate gradient method, it has no minimisation property.

This report is concerned with performance of Bi-CGSTAB on the linear systems which arise during one particular (implicit) numerical solution method for advection-diffusion problems.

A brief history of Bi-CGSTAB is given and relevant convergence theory is reviewed. In numerical experiments, Bi-CGSTAB is found to be as effective as the popular generalized minimum residual method (GMRES) for a relatively mild test case with a moderate convergence criterion, but in a harsher test, the robustness of Bi-CGSTAB is shown to be suspect - rounding errors appear to corrupt the underlying recursion process and lead to divergence.

Various methods are used to improve the robustness of Bi-CGSTAB. Restarting (i.e. discarding the current Krylov subspace and using the latest iterate as a new initial iterate) is found to be the most effective of these methods because it controls the build-up of rounding errors in the recursion process. However, in the absence of an effectively normalised restart criterion, it has the unattractive feature of requiring an external parameter.

Contents

1 Introduction

This report is concerned with the performance of the Bi-CGSTAB nonsymmetric solver on the linear systems which arise during one particular numerical solution method for advection-diffusion problems.

Many discretisation techniques for problems involving advection give rise to large, sparse, symmetric, positive definite matrix systems. These are ideally suited to solution by the conjugate gradient (CG) method which has the advantages that,

1. the only reference to the matrix in the system is a matrix-vector product,
2. the iterates are computed efficiently due to a three-term recurrence,
3. a bound on the error exists which guarantees monotonic convergence and,
4. unlike many other iterative methods (e.g. Chebychev acceleration [8]), no external parameters are required.

Many of the discretisation methods for advection-diffusion problems that lead to symmetric systems treat the advection explicitly, giving rise to either a stability restriction or decreased temporal accuracy (see e.g. Leismann & Frind [13]). If the advection is treated implicitly, there is no formal stability restriction and good temporal accuracy can be obtained but, for a standard Bubnov-Galerkin finite element spatial discretisation, the coefficient matrix in the linear system is nonsymmetric, i.e.

$$A\mathbf{u} = \mathbf{f} \qquad (A^T \neq A) \qquad \text{where } A \in \mathbb{R}^{n\times n} \text{ is invertible and } \mathbf{u}, \mathbf{f} \in \mathbb{R}^n. \tag{1}$$

Due to the nonsymmetry, CG cannot be used to solve (1). But, since most matrices arising from standard spatial discretisation techniques are large and sparse, it is desirable to use a linear solver with properties 1 and 2. This precludes the use of direct methods based on Gaussian elimination except for cases where there is some special structure present. Ideally, the solver should also possess properties 3 and 4. There are many nonsymmetric solvers in the literature but, as yet, none satisfy all of these requirements.

CG may be used to solve (1) if the system is pre-multiplied by A^T to form the normal equations,

$$A^T A\mathbf{u} = A^T\mathbf{f},$$

in which the coefficient matrix is symmetric and positive definite (SPD). There are cases where this approach is optimal (e.g. when A is unitary) but, in general, the coefficient matrix tends to be poorly conditioned and convergence is slow.

Most of the current popular iterative solvers are Krylov subspace methods. At the i^{th} iteration, these attempt to find the solution to (1) in a space $\mathbf{u}_0 + \mathcal{K}_i$ (where $\mathbf{u}_0 \in \mathbb{R}^n$ is the initial iterate and $\mathcal{K}_i$ is a Krylov subspace of dimension i) by imposing the Petrov-Galerkin condition,

$$\mathbf{f} - A\mathbf{u}_i \perp \mathcal{L}_i,$$

where $\mathcal{L}_i$ is another space of dimension i. The class of Krylov subspace methods include CG, the generalized minimal residual (GMRES, Saad and Schultz [19]), quasi-minimal residual (QMR, Freund & Nachtigal [7]) and Bi-CGSTAB (van der Vorst [23]) methods.

There have been many comparative studies on Krylov subspace methods. In Nachtigal *et al* [14], each of a selection of these methods is shown to have the best performance on some carefully constructed examples, and the worst on others. More empirically based studies, where various methods are compared in practical situations (e.g. plasma turbulence modelling [1], groundwater flow [17], semiconductor device modelling [18]), have also shown that the relative performance of these methods depends on the situation.

Since so many comparisons of the different methods exist then, apart from some brief results for GMRES, Bi-CGSTAB is investigated in isolation in this work and the reader is left to use the available literature (e.g. the references cited in the previous paragraph) for general comparison purposes.

This report contains numerical results on the behaviour of Bi-CGSTAB when applied to the system arising from the discretisation of an advection-diffusion equation by the implicit Taylor-Galerkin method. This approach is used in [15] for transport problems arising in hydrology.

The next section contains a brief history of the development of Bi-CGSTAB, a review of some theoretical convergence results for Krylov subspace methods and a statement of the Bi-CGSTAB algorithm. Section 3 describes the nonsymmetric system used to test the solver in this work. Section 4 contains results on the performance of Bi-CGSTAB from numerical experiments under both moderate and harsh convergence criteria - the latter results indicate that rounding errors can lead to divergence. Some techniques for improving convergence behaviour are reviewed in Section 5 and a selection of these are applied to the harsh convergence criterion tests to overcome the divergence problems and improve the robustness of the Bi-CGSTAB solver.

2 Bi-CGSTAB

Bi-CGSTAB is an iterative method for approximating the solution to a linear system in which the matrix is nonsymmetric. It is a bi-orthogonalisation Krylov subspace method, i.e. given an initial iterate $\mathbf{u}_0 \in \mathbb{R}^n$, at the i^{th} iteration the method looks for an approximate solution $\mathbf{u}_i$ of (1) from a space,

$$\mathbf{u}_0 + \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{i-1}\mathbf{r}_0\},$$

(where $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$) by imposing the orthogonalisation,

$$\mathbf{r}_i \perp \{\mathbf{r}_0, (A^T)\mathbf{r}_0, (A^T)^2\mathbf{r}_0, \ldots, (A^T)^{i-1}\mathbf{r}_0\}.$$

In this section, the origin of Bi-CGSTAB is described. This is followed by an overview of convergence results for Krylov subspace methods, and a description of the particular form of the Bi-CGSTAB algorithm used.

2.1 Origin

In the CG method for SPD matrices, the residuals are mutually orthogonal, i.e.

$$(\mathbf{r}_i, \mathbf{r}_j) = 0 \quad (i \neq j),$$

and can be shown to satisfy the expression,

$$\mathbf{r}_i = \psi_i(A)\mathbf{r}_0,$$

where $\psi_i(\cdot)$ is a polynomial of degree $\leq i$ (and $\psi_i(0) = 1$). A three-term recurrence relationship exists between these residuals, and it is the exploitation of this property that makes the algorithm economical in terms of both computing time and storage. This three-term recurrence relation arises from the close relationship of the algorithm with the symmetric Lanczos tridiagonalisation process (see e.g. Golub & Van Loan [8] for a description of this relationship).

When A is nonsymmetric, it is not possible to construct a three-term recurrence relationship of the type used in CG for the residuals. In this case, a possible approach is to base the solver on the *nonsymmetric* Lanczos tridiagonalisation process - this is the origin of the earliest predecessor of Bi-CGSTAB, the bi-conjugate gradient method (Bi-CG) [5, 12].

In Bi-CG, the approximations are constructed in such a way that the residual vector, $\mathbf{r}_i$, is orthogonal to a set of pseudo-residual vectors $\{\hat{\mathbf{r}}_j\}_{(j=1,\ldots,i-1)}$ and, vice versa, $\hat{\mathbf{r}}_i \perp \{\mathbf{r}_j\}_{(j=1,\ldots,i-1)}$. This is accomplished by *two* three-term recurrence relationships for the rows $\mathbf{r}_i$ and $\hat{\mathbf{r}}_i$. The Bi-CG residual vectors are given by,

$$\mathbf{r}_i = \varphi_i(A)\mathbf{r}_0 \quad , \quad \hat{\mathbf{r}}_i = \varphi_i(A^T)\hat{\mathbf{r}}_0,$$

where $\varphi_i(\cdot)$ is a polynomial of degree $\leq i$.

In the case of convergence, both these vectors tend towards zero but only the convergence of $\mathbf{r}_i$ is exploited, $\hat{\mathbf{r}}_j$ only being required for the calculation of iteration parameters. The iteration parameters in Bi-CG are generated by inner products such as the bi-orthogonality condition,

$$(\mathbf{r}_i, \hat{\mathbf{r}}_j) = (\varphi_i(A)\mathbf{r}_0, \varphi_j(A^T)\hat{\mathbf{r}}_0) = 0 \quad (i \neq j). \tag{2}$$

This inner product can be written as,

$$(\mathbf{r}_i, \hat{\mathbf{r}}_j) = (\varphi_j(A)\varphi_i(A)\mathbf{r}_0, \hat{\mathbf{r}}_0) = 0 \quad (i \neq j). \tag{3}$$

Bi-CG was superseded by a faster converging variant - the conjugate gradient-squared method (CG-S) of Sonneveld [21]. In CG-S, all the convergence effort is directed to $\mathbf{r}_i$ by constructing the iteration parameters using the bi-orthogonality condition in the form (3). In this way, the $\hat{\mathbf{r}}_j$ do not need to be formed and there is no requirement for A^T.

The CG-S residual vectors are given by,

$$\mathbf{r}_i = \varphi_i^2(A)\mathbf{r}_0,$$

so, if $\varphi_i(A)$ is viewed as a Bi-CG contraction operator (in the case of convergence), then the CG-S contraction operator, $\varphi_i^2(A)$, is twice as effective. However, situations arise in the iteration process where $\varphi_i(A)$ is not a contraction operator and spikes occur in the convergence history. In practice CG-S is found to have a rather erratic convergence behaviour, particularly when the starting iterate is close to the solution (as is the case in many transient and non-linear problems). This tends not to slow the overall convergence rate of the method but, in finite precision, it can allow rounding errors to cause a breakdown in the method.

Bi-CGSTAB is a variant of CG-S which has a more smoothly varying convergence history. This method comes from a generalisation of the Bi-CG and CG-S methods; the orthogonality conditions (2) and (3) are imposed in those methods but other iteration methods can be generated by choosing,

$$\mathbf{r}_i \perp \left\{\tilde{\varphi}_j(A^T)\hat{\mathbf{r}}_0\right\}_{(j=1,\ldots,i-1)}$$

where $\tilde{\varphi}_j(\cdot)$ is another polynomial of degree $\leq j$, (in Bi-CG, $\tilde{\varphi}_j(\cdot) = \varphi_j(\cdot)$).

There is a degree of freedom in which to choose the polynomial $\tilde{\varphi}_i(\cdot)$ while using inner products in the form (3) but having a more smoothly varying convergence history. A constraint on the choice of this polynomial is that it should allow the use of recursion relationships in the resulting algorithms to keep the computational work and storage requirements small at each iteration.

In Bi-CGSTAB, $\tilde{\varphi}_i(\cdot)$ is of the form,

$$\tilde{\varphi}_i(A) = (1 - \omega_1 A)(1 - \omega_2 A)\ldots(1 - \omega_i A), \tag{4}$$

where ω_j $(j = 1, \ldots, i)$ are constants determined by a local steepest descent step.

2.2 Convergence Theory

In exact arithmetic, Krylov subspace methods have a finite termination property[1], but this is of little practical use.

For the case $(\mathcal{K}_i = \mathcal{L}_i = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{i-1}\mathbf{r}_0\})$, if A is SPD, the resulting Krylov subspace method (CG) possesses the optimality property,

$$\|\mathbf{u} - \mathbf{u}_i\|_A = \min_{\mathbf{w} \in \mathbf{u}_0 + \mathcal{K}_i} \|A^{-1}\mathbf{f} - \mathbf{w}\|_A \tag{5}$$

[1]If a solution exists, the process finds this in at most n iterations.

where $\mathbf{u}$ is the exact solution of the linear system, $\mathbf{u}_i$ is the i^{th} iterate produced by the conjugate gradient method and $\| \cdot \|_A$ is the A-norm defined by,

$$\|\mathbf{w}\|_A = \sqrt{\mathbf{w}^T A \mathbf{w}}.$$

The class of Krylov subspace iteration methods to which GMRES belongs (i.e. $\mathcal{K}_i$ as in CG and $\mathcal{L}_i = A\mathcal{K}_i$) possesses the optimality property,

$$\|\mathbf{r}_i\|_2 = \min_{\mathbf{w} \in \mathbf{u}_0 + \mathcal{K}_i} \|\mathbf{f} - A\mathbf{w}\|_2.$$

The existence of such a property guarantees that the residual is a monotonic function of the iteration number.

Unfortunately, the main theorem from Faber & Manteuffel [4] states that CG-like methods with (i) a minimisation property and (ii) cheap short-term recurrence relationships exist only for special matrices. In general, CG-like methods possess either (i) or (ii) but not both, e.g. GMRES has (i) but can be expensive to implement (the work at each iteration grows linearly) while Bi-CG has (ii) but has no minimisation property.

An apparent exception is the quasi-minimal residual method which has three-term recurrence relationships and also minimises the residual in the norm, $\|D_{i+1}^{-1} W_{i+1}^T \mathbf{r}_i\|_2$, at the i^{th} iteration (where D and W are matrices associated with the underlying nonsymmetric Lanczos process - see [6] for more detail). However, as the name suggests, QMR is not a true minimisation process; the residual is minimised in a norm that changes with each iteration. Because of this, QMR falls out of the scope of the Faber & Manteuffel theorem.

Since it is based on three-term recurrence relationships and possesses no quasi-minimisation property, no convergence bounds exist for Bi-CGSTAB. Current knowledge of the practical behaviour of the method falls into two main categories:

- Investigations on the effects of the presence of extreme (large, small and negative) eigenvalues in the eigenspectrum of the coefficient matrix (e.g. for a comparison of this type with Bi-CG and CG-S, see Campos,filho & Rollet [2]). These studies tend to use matrices constructed to generate a particular eigenspectrum.

- Comparisons with other solvers on matrices arising from the solution of practical problems (e.g. Peters [17]).

The work in this report uses a coefficient matrix whose properties are controlled by the physical values associated with the problem being solved and the size of the temporal and spatial discretisation used in the numerical approximation of the underlying differential equation, and examines the behaviour of the Bi-CGSTAB method in isolation as the parameters are varied.

2.3 Basic Preconditioned Bi-CGSTAB Algorithm

There are many possible forms of the Bi-CGSTAB algorithm which are equivalent in exact arithmetic but which show different behaviour in finite precision. For this reason, the form of the Bi-CGSTAB algorithm used in this work is given in this section.

The following algorithm is similar to the one given by van der Vorst [23], but incorporates some of the modifications suggested in that original paper. This algorithm searches iteratively for a solution to (1) where the matrix is preconditioned by K.

```
$\mathbf{u}$ is an initial iterate; $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}$;
$\rho_0 = \alpha = \omega = 1$;  $\rho_1 = (\mathbf{r}_0, \mathbf{r}_0)$;
$\mathbf{v} = \mathbf{p} = \mathbf{0}$;
do $i = 1, i_{max}$
    $\beta = (\rho_i/\rho_{i-1})(\alpha/\omega)$;                (6)
    $\mathbf{p} = \mathbf{r}_{i-1} + \beta(\mathbf{p} - \omega\mathbf{v})$;                (7)
    Solve $\mathbf{y}$ from $K\mathbf{y} = \mathbf{p}$;
    $\mathbf{v} = A\mathbf{y}$;
    $\alpha = \rho_i/(\mathbf{r}_0, \mathbf{v})$;                (8)
    $\mathbf{s} = \mathbf{r}_{i-1} - \alpha\mathbf{v}$;                (9)
    Solve $\mathbf{z}$ from $K\mathbf{z} = \mathbf{s}$;
    $\mathbf{t} = A\mathbf{z}$;
    $\omega = (\mathbf{t}, \mathbf{s})/(\mathbf{t}, \mathbf{t})$;
    $\rho_{i+1} = -\omega(\mathbf{r}_0, \mathbf{t})$;                (10)
    $\mathbf{u} = \mathbf{u} + \alpha\mathbf{y} + \omega\mathbf{z}$;
    if $\mathbf{u}$ is accurate enough then quit;
    $\mathbf{r}_i = \mathbf{s} - \omega_i\mathbf{t}$;
end do
```

3 Nonsymmetric System Under Consideration

The nonsymmetric system in this work arises from an implicit discretisation of the equation for contaminant transport in porous medium, a full description of the origin of this system being given in [15]. In this section a brief summary of the relevant parts of that work is presented; this includes a description of the partial differential equation being approximated, the discretisation used and a simple test case arising from the solution of a 1-D problem.

3.1 Governing Equation for Contaminant Transport in Porous Media

From [15], the mass balance equation for a contaminant in a saturated porous medium is,

$$\rho\phi\frac{\partial c}{\partial t} + (\rho\mathbf{q}).\nabla c = \nabla.\left\{\phi\underline{\underline{\mathbf{D}}}\nabla(\rho c)\right\}, \tag{11}$$

where $\rho = \rho(c)$ is fluid density , ϕ is the porosity, c is the (dimensionless) contaminant concentration, $\mathbf{q}$ is the Darcy velocity and $\underline{\underline{\mathbf{D}}}$ is the dispersion tensor.

In this work, ρ, ϕ, $\underline{\underline{\mathbf{D}}}$ and $\mathbf{q}$ are taken to be constant. With these simplifications (11) becomes the constant coefficient advection-diffusion equation,

$$\frac{\partial c}{\partial t} + \mathbf{v}.\nabla c = \nabla.\left(\underline{\underline{\mathbf{D}}}\nabla c\right), \tag{12}$$

where $\mathbf{v} = \dfrac{\mathbf{q}}{\phi}$ is the average fluid velocity.

3.2 Discretisation of the Governing Equation

The advection-diffusion equation representing the contaminant mass balance is discretised by an implicit Taylor-Galerkin method - using (12) to replace the temporal derivatives in an approximate Taylor series expansion, and then performing a spatial discretisation by the standard Galerkin finite element method. This gives the same result as the Crank-Nicolson finite element method [17].

The fully discretised form of the contaminant mass balance equation is,

$$\left\{\frac{1}{\Delta t}A + \frac{1}{2}(B + C)\right\}\mathbf{c}^{t+\Delta t} = \left\{\frac{1}{\Delta t}A - \frac{1}{2}(B + C)\right\}\mathbf{c}^{t} - \mathbf{F}, \tag{13}$$

where

$$\begin{aligned} A &= \{A_{IJ}\}_{I,J=1,\ldots,n} & & \\ B &= \{B_{IJ}\}_{I,J=1,\ldots,n} & \mathbf{c}^i &= \{c^i_J\}_{J=1,\ldots,n} \\ C &= \{C_{IJ}\}_{I,J=1,\ldots,n} & \mathbf{F} &= \{F_I\}_{I=1,\ldots,n}, \end{aligned}$$

and

$$\begin{aligned} A_{IJ} &= \sum_e A^e_{IJ} = \sum_e \int_{\Omega^e} N_I N_J d\Omega^e \\ B_{IJ} &= \sum_e B^e_{IJ} = \sum_e \int_{\Omega^e} \nabla N_I.\underline{\underline{\mathbf{D}}}\nabla N_J d\Omega^e \end{aligned}$$

$$\begin{aligned} C_{IJ} &= \sum_e C^e_{IJ} = \sum_e \mathbf{v}. \int_{\Omega^e} N_I \nabla N_J d\Omega^e \\ F_I &= \sum_e F^e_I = \sum_e \frac{q^c_n}{\phi} \int_{\Gamma^e} N_I d\Gamma^e, \end{aligned}$$

where the superscript, e, denotes an element value and $\sum_e$ denotes summation over all the elements.

3.3 General Properties of the Coefficient Matrix

The coefficient matrix in (13) is a combination of the finite element mass matrix A, stiffness matrix B, and advection matrix C. The mass matrix is SPD. After applying Dirichlet boundary conditions, the stiffness matrix is SPD if $\underline{\underline{\mathbf{D}}}$ is SPD (Davies [3], p212). The advection matrix is non-symmetric and indefinite.

The proportion of each of these matrices (and hence the properties of the full matrix) depends on the relative sizes of the finite elements, Δt, $\underline{\underline{\mathbf{D}}}$ and $\mathbf{v}$. There are two important dimensionless parameters which characterise advection-diffusion flows : the Courant number (ν) which gives an indication of the relative importance of advection and inertia (i.e. the proportion of the advection matrix to the mass matrix), and the mesh Peclet number (Pe) which gives the relative importance of advection and diffusion (i.e. the proportion of the advection matrix to the stiffness matrix).

3.4 Test Problem : Transport of Tracer in a Column

This problem involves the 1-D transport of a tracer in a vertical column through which there is a constant fluid flow rate.

The physical region is a rectangle which is 2000m high (and 10m wide), details of the boundary conditions for which can be found in [15]. The average fluid velocity vector and dispersion tensor are,

$$\mathbf{v} = \begin{bmatrix} 0 \\ -0.00005 \end{bmatrix} \text{m/s} \quad , \quad \underline{\underline{\mathbf{D}}} = \begin{bmatrix} 0 & 0 \\ 0 & 0.00025 \end{bmatrix} \text{m}^2/\text{s}.$$

Initially, the concentration of the tracer is zero everywhere inside the region. The tracer front moves down the column under the action of gravity and this front disperses as it moves.

The Courant and mesh Peclet numbers for the problem are,

$$\nu = (5 \times 10^{-5}) \frac{\Delta t}{\Delta z} \qquad Pe = 0.2\Delta z,$$

where Δt is the time step and Δz is the vertical mesh size. Two different grids of uniform bilinear rectangular elements are used to give results at different mesh Peclet numbers. The first grid has 401 nodes in the z-direction (and 2 nodes in the x-direction) which corresponds to a mesh Peclet number of $Pe = 1$, and the second grid has 81 nodes in the z-direction (and 2 nodes in the x-direction) which corresponds to a mesh Peclet number of $Pe = 5$. The problems on these grids gives rise to matrices of dimension 802 and 162 respectively.

4 Performance of Bi-CGSTAB

In this section, the performance of Bi-CGSTAB on the linear systems arising from the test case in Section 3 is examined. As with CG, preconditioning is an important aspect of the behaviour of this method. However, this is not considered here - no preconditioning (i.e. $K = I$) is used in the moderate convergence tests (to allow a comparison with GMRES) and diagonal preconditioning (i.e. $K = \mathrm{diag}(A)$) is used the harsh convergence tests.

All the numerical experiments are carried out in double precision on a Sun SPARC-station 1+ workstation. The round-off unit, ϵ, is used intermittently in these experiments. It is defined as the largest number which satisfies,

$$fl(1+\epsilon) = 1. \tag{14}$$

For the f77 compiler on the machine used, $\epsilon \approx 2.22045 \times 10^{-16}$.

The experiments fall into two groups, (i) moderate convergence tests in which the linear solver is given a practical convergence tolerance for reasonable physical problems, and (ii) harsh convergence tests where the solver is required to converge to the machine accuracy on problems where the physical parameters are given more extreme values. The first set of tests examine the practical efficiency of the solver while the second set of tests examine its robustness.

4.1 Moderate Convergence Criterion

In this section, convergence is taken to be,

$$\|\mathbf{r}_i\|_2 \leq (tol)\|\mathbf{f}\|_2, \tag{15}$$

where $tol = 10^{-8}$.

The test case is run from the initial condition to a fixed time ($t = 2.25 \times 10^7$ seconds) with different time steps used to give different Courant numbers. Table 1 shows the performance of the algorithm under these conditions.

Pe	n	ν	Average iterations per time step		Total CPU / sec	
			Bi-CGSTAB	GMRES	Bi-CGSTAB	GMRES
1	802	0.5	1.79	2.75	710.9	690.2
		1	2.30	4.37	367.2	364.7
		5	12.98	23.71	124.5	162.0
5	162	0.5	2.54	4.33	29.7	29.1
		1	3.91	6.13	16.1	15.9
		5	19.44	27.33	6.1	7.6

Table 1: Moderate convergence criterion performance

The average number of iterations for convergence increases with the Courant number, i.e. as the mass matrix becomes less dominant. However, due to the larger time-steps used at the higher Courant number, the overall running time of the test case is reduced

- this is the reason for the use of an implicit discretisation (as opposed to an explicit one which has a time step restriction).

The performance of GMRES (with the same preconditioner and stopping criterion) on the same problem is also shown in Table 1. Bi-CGSTAB requires approximately half as many iterations per time step as GMRES, but each Bi-CGSTAB iteration contains two matrix-vector products compared to the one required by GMRES, hence the CPU times of the two methods are similar.

Bi-CGSTAB can be seen to become more efficient as the number of iterations required for convergence increases (since the work per iteration grows linearly for GMRES). However, Bi-CGSTAB does not have the stable, monotonic convergence properties of GMRES (or the CG solver that would be used in the symmetric systems resulting from an explicit discretisation). For this reason, there is a need to examine the performance of Bi-CGSTAB under harsher conditions.

4.2 Harsh Convergence Criterion

In this section, an extreme tolerance is requested in (15), i.e. $tol = \epsilon$ (the round-off unit), and larger Courant numbers are used (increasing the nonsymmetry).

Due to the possibility of non-convergence in the linear solver, these tests are only run for a single time step. Also, to reduce the effect of initial stiffness in the test case, the tests are started from time $t_0 = 7.5 \times 10^6$ seconds (the initial profile being generated by an analytic solution).

Pe	ν	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Iteration
1	0.5	$< \epsilon$	8
	1	4.86×10^{-16}	12
	2	4.91×10^{-16}	20
	5	5.32×10^{-14}	44
	10	3.34×10^{-13}	56
	20	1.89×10^{-10}	70
	40	4.93×10^{-8}	77
5	0.5	6.90×10^{-14}	8
	1	1.16×10^{-15}	13
	2	5.47×10^{-16}	20
	5	1.67×10^{-12}	35
	10	2.60×10^{-8}	41
	20	4.70×10^{-7}	45
	40	9.59×10^{-6}	49

Table 2: Harsh convergence criterion performance of Bi-CGSTAB

Table 2 shows the minimum residual which occurs during the convergence history, and the corresponding iteration number. Convergence is only achieved in one of the cases shown. In half the cases the minimum residual is not within three orders of magnitude of convergence.

Figures 1 and 2 are the convergence histories for two of the cases in Table 2, the horizontal line near -15.5 is the convergence target. In both figures, convergence is steady and reasonably fast in the early stages and the residual norm decreases almost monotonically. But a worrying feature is that the good convergence behaviour stops at some stage and the residual norm diverges (i.e. shows a general trend of increasing) - this is representative of the behaviour in most of the harsh convergence tests.

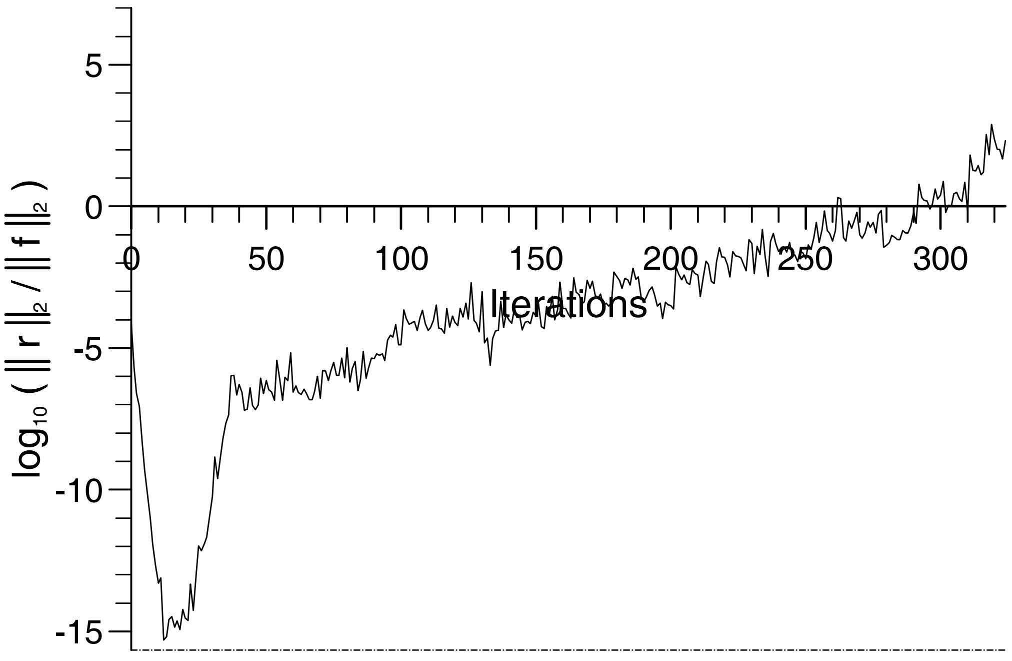

Figure 1: Convergence history for $\nu = 1$, $Pe = 1$

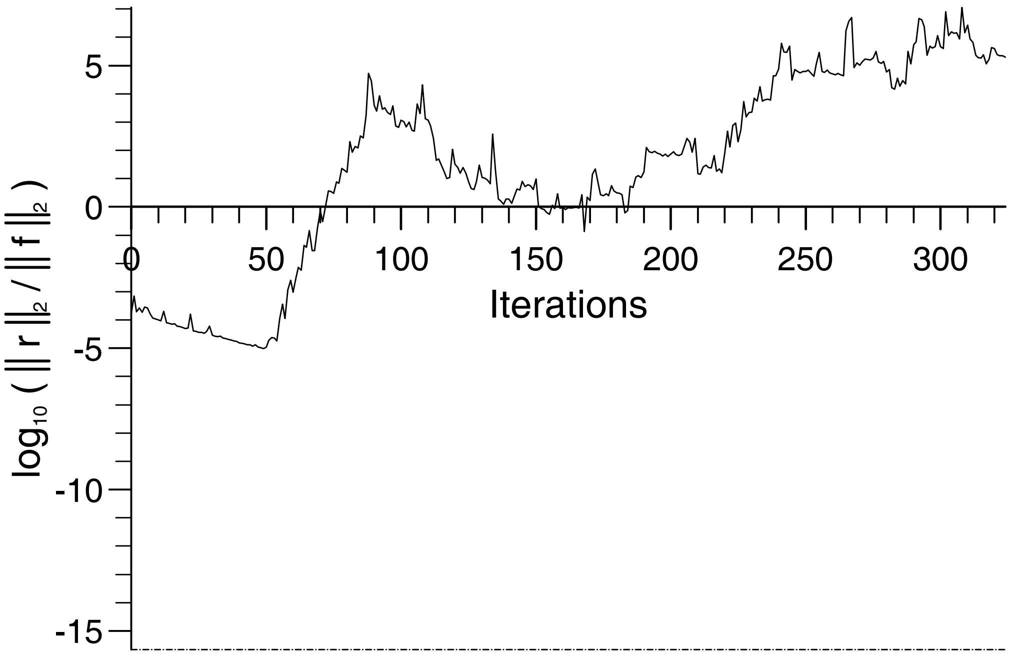

Figure 2: Convergence history for $\nu = 40$, $Pe = 5$

It is possible that the divergence is caused by the rounding errors which occur in finite precision arithmetic. Table 3 shows the maximum relative error between the (recursion-based) Bi-CGSTAB residuals and the true residuals (calculated by $\mathbf{r}_i^{true} = \mathbf{f} - A\mathbf{u}_i$), and also the iteration when this maximum occurs. These errors are quite small, so a total breakdown of the recursion process has not occurred; but the small discrepancy between the true and recursion residuals may indicate that the recursion process has been spoiled, leading to the convergence difficulties. Indeed, comparing Tables 2 and 3, the onset of divergence often coincides with the largest relative error in the residual.

Pe	ν	$\max_i \frac{\|\mathbf{r}_i^{true} - \mathbf{r}_i\|_2}{\|\mathbf{r}_i^{true}\|_2}$	Iteration
1	0.5	8.22×10^{-3}	8
	1	2.51×10^{-3}	12
	2	2.54×10^{-3}	20
	5	6.49×10^{-5}	36
	10	5.24×10^{-6}	56
	20	1.27×10^{-8}	70
	40	3.83×10^{-8}	375
5	0.5	7.04×10^{-6}	8
	1	4.04×10^{-4}	13
	2	7.60×10^{-4}	20
	5	1.94×10^{-7}	35
	10	1.32×10^{-10}	39
	20	3.29×10^{-11}	147
	40	1.09×10^{-10}	168

Table 3: Relative error in residuals

If Bi-CGSTAB is to be considered as a viable alternative to other nonsymmetric solvers for this problem, the convergence difficulties highlighted by the harsh convergence criterion tests must be overcome so that the solver is robust as well as efficient.

5 Improving the Performance of Bi-CGSTAB

In this section, an attempt is made to prevent the divergent behaviour in the harsh convergence criterion tests highlighted by Figures 1 and 2. Some techniques for improving convergence behaviour currently in the literature are:

- **look-ahead Lanczos** - it is known that the Lanczos tridiagonalisation process (which underpins three-term recurrence relationship methods) is unstable and prone to breakdown due to a loss of orthogonality between the Lanczos vectors [8]. To remedy this, the look-ahead Lanczos process of Parlett *et al* [16] allows the use of block pivots in the iteration steps where the scalar pivots of the standard Lanczos process is expected to encounter difficulties. This method is used in practical versions of QMR.

- variants of Bi-CGSTAB (e.g. Bi-CGSTAB2 [9], Bi-CGSTAB(ℓ) [20]) allow **higher order polynomials** in the construction of $\tilde{\varphi}_i(A)$ in (4) rather than the linear components, $(1 - \omega_j A)$, used in the original van der Vorst version. These methods attempt to avoid stagnation in the convergence history of Bi-CGSTAB which occurs when the eigenvalues are almost purely imaginary.

- **residual smoothing** (Weiss & Schönauer [24]) - an auxiliary sequence of vectors, $\bar{\mathbf{u}}_i$, is generated from non-monotonic iterates, $\mathbf{u}_i$, by the recursion,

  $$\begin{aligned} \bar{\mathbf{u}}_0 &= \mathbf{u}_0 \\ \bar{\mathbf{u}}_i &= (1-\eta_i)\bar{\mathbf{u}}_{i-1} + \eta_i \mathbf{u}_i \quad (i = 1, 2, \ldots), \end{aligned}$$

  where each η_i is chosen to minimise,

  $$\|\mathbf{f} - A\left\{(1-\eta)\bar{\mathbf{u}}_{i-1} + \eta \mathbf{u}_i\right\}\|_2,$$

  over $\eta \in \mathbb{R}$. η_i is given explicitly by,

  $$\eta_i = -\frac{\mathbf{s}_{i-1}^T(\mathbf{r}_i - \mathbf{s}_{i-1})}{\|\mathbf{r}_i - \mathbf{s}_{i-1}\|_2^2},$$

  where $\mathbf{s}_{i-1} = \mathbf{f} - A\bar{\mathbf{u}}_{i-1}$. The vectors in the auxiliary sequence, $\bar{\mathbf{u}}_i$, are iterates with monotone non-increasing residual.

- **random initial iterate** - the Mismatch Theorem [22] indicates that a breakdown in the Lanczos process can be caused by "irregular" left- and right-eigenvector distributions in $\mathbf{r}_0$. Joubert [11] suggests that this problem can be overcome by using an appropriately scaled initial vector consisting of random entries. Since good initial iterates are often available in time-dependent and non-linear problems, it is more suitable in these cases to add a perturbation to the initial vector.

- **restarting** - if the recursion process is spoiled by rounding errors then, by restarting the iteration with a new initial iterate (e.g. the latest one), the current numerical Krylov subspace is discarded and with it all the rounding errors to date. This method is advocated (although for other reasons) for Bi-CGSTAB by van der Vorst [23].

As the convergence problems are not caused by breakdown or stagnation, the look-ahead Lanczos and higher order polynomial methods are not investigated here.

5.1 Residual Smoothing

Rather than the residual smoothing algorithm already described, the following one (from [25]), which is the same in exact arithmetic but has better numerical rounding properties, is used here.

```
$\bar{\mathbf{r}}_0 = \mathbf{r}_0$ ; $\bar{\mathbf{u}}_0 = \mathbf{u}_0$ ; $\mathbf{dr} = \mathbf{du} = \mathbf{0}$;
For $i = 1, 2, \ldots$
    Generate $\alpha$, $\omega$, $\mathbf{t}$, $\mathbf{v}$, $\mathbf{y}$ and $\mathbf{z}$ by a Bi-CGSTAB iteration ;
    $\mathbf{dr} = \mathbf{dr} + \alpha\mathbf{v} + \omega\mathbf{t}$ ; $\mathbf{du} = \mathbf{du} + \alpha\mathbf{y} + \omega\mathbf{z}$ ;
    $\eta = \frac{(\bar{\mathbf{r}}_{i-1}, \mathbf{dr})}{(\mathbf{dr}, \mathbf{dr})}$ ;
    $\bar{\mathbf{r}}_i = \bar{\mathbf{r}}_{i-1} - \eta\mathbf{dr}$ ; $\bar{\mathbf{u}}_i = \bar{\mathbf{u}}_{i-1} + \eta\mathbf{du}$ ;
    $\mathbf{du} = (1-\eta)\mathbf{du}$ ; $\mathbf{dr} = (1-\eta)\mathbf{dr}$ ;
```

Table 4 is the residual smoothed equivalent of Table 2, it shows the the minimum residual and the iteration at which this was reached.

Pe	ν	$(\|\bar{\mathbf{r}}\|_2/\|\mathbf{f}\|_2)_{min}$	Iteration
1	0.5	$< \epsilon$	8
	1	4.07×10^{-16}	27
	2	3.46×10^{-16}	48
	5	1.01×10^{-14}	55
	10	1.83×10^{-13}	68
	20	1.17×10^{-10}	91
	40	2.54×10^{-8}	131
5	0.5	4.72×10^{-14}	23
	1	3.87×10^{-16}	130
	2	2.86×10^{-16}	29
	5	5.25×10^{-13}	44
	10	2.09×10^{-9}	49
	20	2.26×10^{-7}	56
	40	5.13×10^{-6}	61

Table 4: Harsh convergence performance with residual smoothing

Residual smoothing produces residuals that are lower than the Bi-CGSTAB residuals from which they are generated. However, in general, the improvement is not enough to warrant even the small amount of extra work involved.

Residual smoothing does not improve convergence significantly because the original Bi-CGSTAB iterates on which the smoothing is based are still prone to the same problems as before, hence the residual smoothing stagnates (i.e. $\eta = 0$) when Bi-CGSTAB stops converging.

An obvious advantage of the process is that the solution is monotone non-increasing, however the effect is only marginally better than the cheaper method of storing the best current iterate. In the cases presented, residual smoothing is of little use in improving the performance under harsh convergence conditions.

5.2 Random Initial Iterate

Since a good initial iterate, $\mathbf{u}_0$, is available, a vector of random entries ($\mathbf{V}$) is added to this to generate a new initial iterate $\tilde{\mathbf{u}}_0$. The size of the random perturbation is controlled by scaling and a factor, τ, i.e.

$$\tilde{\mathbf{u}}_0 = \mathbf{u}_0 + \tau \|\mathbf{u}_0\|_2 \frac{\mathbf{V}}{\|\mathbf{V}\|_2} \quad \text{with} \quad \mathbf{V} = \{V_i\}_{i=1,\ldots,n} \quad \text{and} \quad -1 \leq V_i \leq 1.$$

Table 5 shows the minimum residual achieved and the corresponding iteration number for the harsh convergence tests with different perturbation sizes.

Pe	ν	$\tau = 10^{-3}$		$\tau = 10^{-2}$		$\tau = 10^{-1}$	
		$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.
1	0.5	2.80×10^{-16}	20	1.10×10^{-15}	20	1.31×10^{-14}	17
	1	$< \epsilon$	25	5.03×10^{-16}	27	1.90×10^{-14}	25
	2	$< \epsilon$	50	$< \epsilon$	42	7.88×10^{-14}	41
	5	$< \epsilon$	89	1.12×10^{-14}	79	6.72×10^{-14}	69
	10	1.96×10^{-15}	129	2.59×10^{-14}	116	2.31×10^{-13}	138
	20	3.16×10^{-15}	400	5.23×10^{-14}	414	3.53×10^{-13}	224
	40	7.70×10^{-15}	413	2.36×10^{-15}	431	8.94×10^{-14}	469
5	0.5	$< \epsilon$	22	2.01×10^{-15}	24	1.51×10^{-14}	25
	1	2.18×10^{-15}	29	3.89×10^{-15}	25	1.46×10^{-14}	25
	2	4.98×10^{-15}	30	4.10×10^{-14}	30	5.83×10^{-13}	28
	5	1.55×10^{-14}	72	1.15×10^{-13}	71	2.23×10^{-13}	107
	10	7.18×10^{-14}	135	6.48×10^{-14}	157	1.02×10^{-12}	149
	20	3.74×10^{-15}	258	2.50×10^{-13}	243	8.77×10^{-13}	239
	40	4.18×10^{-15}	287	1.21×10^{-13}	292	6.96×10^{-13}	302

Table 5: Harsh convergence performance with random initial iterate

Perturbing the initial vector improves the robustness of the Bi-CGSTAB iteration in the sense that the minimum residual is closer to the harsh convergence criterion over the range of the tests (c.f. Table 2).

Additionally, in many cases (particularly at $Pe = 1$) the convergence history does not diverge after the minimum is reached, but stagnates instead.

The most effective perturbation is the smallest, $\tau = 10^{-3}$ (i.e. 0.1 %) - increasing the size of the perturbation appears to have a detrimental effect.

5.3 Restarting

In the original Bi-CGSTAB paper [23], van der Vorst recommends that practical implementations of the method should monitor sensitive values and, if any of these become too small, the iteration should be stopped and Bi-CGSTAB should be restarted with a new choice of initial iterate. The suggested sensitive values are the scalars ρ_{i+1} and

$(\mathbf{r}_0, \mathbf{v})$ (see Section 2.3), which are highlighted because they may be close to zero without convergence having taken place, and are used as denominators in the algorithm (so are particularly sensitive to round-off error and may cause breakdown by division by zero).

Due to the cost of calculating the initial residual and initialising vectors, a restart requires about half the computational effort of a Bi-CGSTAB iteration. Restarting is often used in the literature but the criteria tend to be heuristic and of the form,

$$\text{Restart if monitor} < \text{tolerance.}$$

Often no guidance is given on the origin of the monitor or the sensitivity to the tolerance.

If the restart criterion is not severe enough, then restart is premature and good convergence behaviour can be interrupted and spoiled. This is similar to the problem which can arise in the practical implementation of GMRES (GMRES(k)) which, in order to decrease the storage requirements and average work per iteration, restarts after every k iterations. If the restart criterion is too lenient, rounding errors can build up and cause the divergent behaviour shown in Figures 1 and 2. In this section, various restart criteria are examined.

5.3.1 Fixed Period Restart

The first criterion investigated is one in which the process is restarted periodically after a fixed number of iterations. This is similar to GMRES(k) but, where a restart is performed to avoid impractical storage requirements and work per iteration in that case, here it is performed at regular intervals to try to prevent the build-up of rounding errors spoiling the recursion process.

Table 6 shows the minimum residual achieved and the corresponding iteration number for the harsh convergence tests with different fixed restart periods, k. The harsh convergence criterion is met in all the tests.

Pe	ν	$k=5$		$k=20$		$k=40$	
		$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Itn.
1	0.5	$<\epsilon$	9	$<\epsilon$	8	$<\epsilon$	8
	1	$<\epsilon$	15	$<\epsilon$	23	$<\epsilon$	61
	2	$<\epsilon$	27	$<\epsilon$	22	$<\epsilon$	51
	5	$<\epsilon$	50	$<\epsilon$	44	$<\epsilon$	53
	10	$<\epsilon$	94	$<\epsilon$	81	$<\epsilon$	78
	20	$<\epsilon$	233	$<\epsilon$	153	$<\epsilon$	149
	40	$<\epsilon$	268	$<\epsilon$	242	$<\epsilon$	193
5	0.5	$<\epsilon$	13	$<\epsilon$	36	$<\epsilon$	54
	1	$<\epsilon$	20	$<\epsilon$	23	$<\epsilon$	59
	2	$<\epsilon$	22	$<\epsilon$	23	$<\epsilon$	50
	5	$<\epsilon$	52	$<\epsilon$	51	$<\epsilon$	64
	10	$<\epsilon$	92	$<\epsilon$	67	$<\epsilon$	84
	20	$<\epsilon$	122	$<\epsilon$	102	$<\epsilon$	88
	40	$<\epsilon$	167	$<\epsilon$	121	$<\epsilon$	92

Table 6: Harsh convergence performance with fixed restart period

Figures 3 and 4 correspond to Figures 1 and 2 for a restart of fixed period $k = 20$.

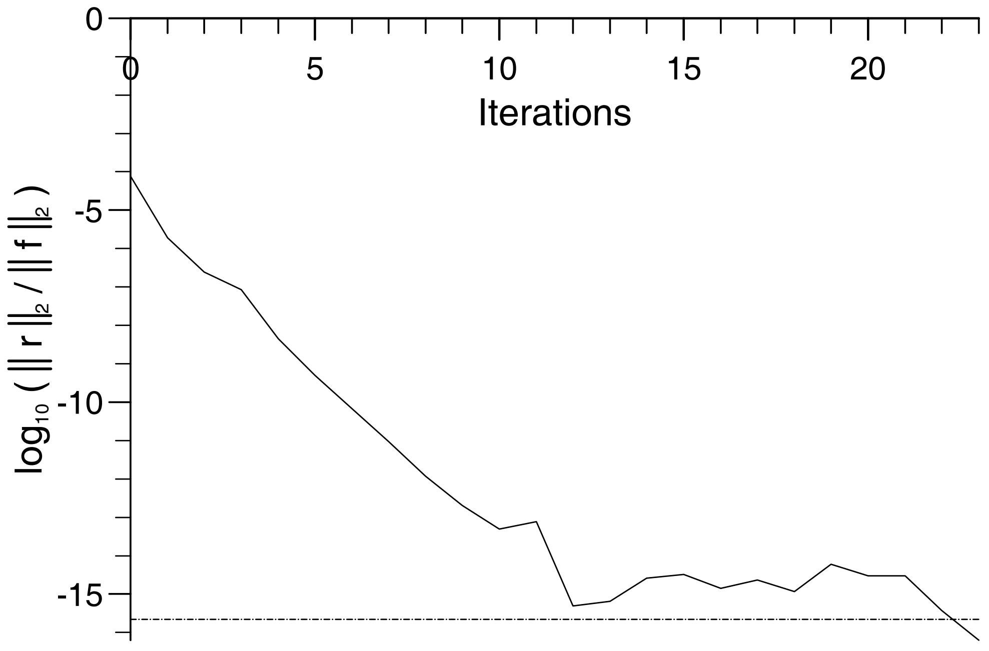

Figure 3: Fixed restart of period $k = 20$ for $\nu = 1$, $Pe = 1$

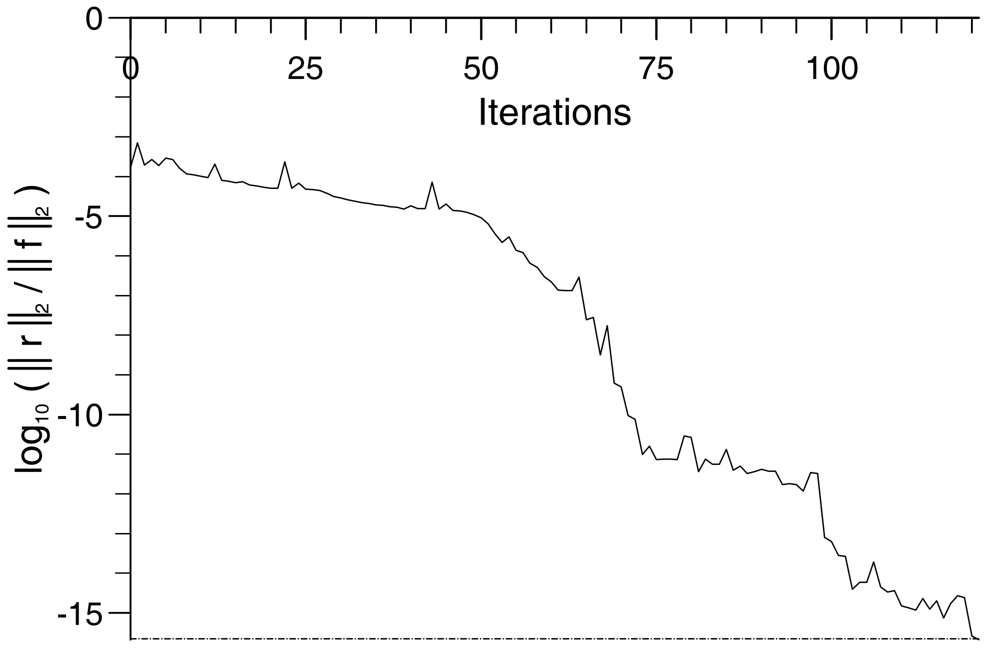

Figure 4: Fixed restart of period $k = 20$ for $\nu = 40$, $Pe = 5$

The value of the restart period k has a significant effect on the number of iterations required to achieve convergence - practical implementations of a fixed restart period should choose k based on the expected susceptibility of the recursion process to corruption by rounding errors.

Generally, larger linear systems are more susceptible to rounding errors due to the number of computational operations involved in each iteration. From Table 6, the optimum restart period decreases as ν is increased, indicating that the more nonsymmetric the matrix, the more sensitive the process is to rounding errors. Thus, the optimum value of k for this problem is expected to decrease with n, ν and Pe (the latter is included because it also affects the amount of nonsymmetry in the matrix).

The requirement for the value k violates the desired property that the solver should require no external parameters (see Section 1).

5.3.2 Restart monitors based on round-off unit definition

From the operation in the Bi-CGSTAB algorithm where ρ is used as a denominator (got by combining (6) and (7)),

$$\mathbf{p} = \mathbf{r}_{i-1} + \frac{\rho_i}{\rho_{i-1}} \frac{\alpha}{\omega} (\mathbf{p} - \omega \mathbf{v}).$$

and the definition of the round-off unit (14), rounding errors can be expected to occur in this operation if,

$$\epsilon > \left| \frac{\rho_i}{\rho_{i-1}} \frac{\alpha}{\omega} \frac{\{\mathbf{p}(j) - \omega \mathbf{v}(j)\}}{\mathbf{r}_{i-1}(j)} \right| > \frac{1}{\epsilon} \quad (1 \leq j \leq n), \tag{16}$$

where $\mathbf{v}(j)$ is the j^{th} component of $\mathbf{v}$. The expression is undefined when a term in the residual vector is zero so these cases must be excluded. Hence a more suitable form of (16) is

$$\mathbf{r}_{i-1}(j) \neq 0 \;\text{ and }\; \epsilon |\mathbf{r}_{i-1}(j)| > |\beta \{\mathbf{p}(j) - \omega \mathbf{v}(j)\}| > \frac{1}{\epsilon} |\mathbf{r}_{i-1}(j)| \quad (1 \leq j \leq n), \tag{17}$$

which is a possible restart criterion, requiring no external parameters other than the easily available round-off unit. However, experimental investigations show that in all the harsh convergence criterion tests, the monitor value is always well within the bounds which indicate round-off and there is no interesting behaviour in the monitor near the onset of convergence difficulties - hence (17) is of no use for this problem.

In the same way as round-off error in ρ_{i+1} is tested for in (17), round-off can be tested for in $(\mathbf{r}_0, \mathbf{v})$ by combining (8) and (9) and using the definition of the round-off unit to give the possible restart criterion,

$$\mathbf{r}_{i-1}(j) \neq 0 \;\text{ and }\; \epsilon |\mathbf{r}_{i-1}(j)| > |\alpha \mathbf{v}(j)| > \frac{1}{\epsilon} |\mathbf{r}_{i-1}(j)| \quad (1 \leq j \leq n). \tag{18}$$

Again, experimental investigations show that this monitor is always well within the bounds for all the tests, and there are no significant features in the monitor near the onset of convergence difficulties, so this restart criterion is also of no use.

5.3.3 Restart monitors based on inner product denominators

Joubert [10] uses restart criteria based on the occurrence of inner products as denominators in Bi-CG and CG-S. The general form of these criteria are that if $(\mathbf{a}, A\mathbf{b})$ is used as a denominator then restart if,

$$\frac{|(\mathbf{a}, A\mathbf{b})|}{\|\mathbf{a}\|_2 \|A\mathbf{b}\|_2} \leq f(\epsilon).$$

In [10], the tolerance has the form, $f(\epsilon) = 10^a \epsilon^{\frac{1}{2}}$, where a is an integer.

Since the inner product $(\mathbf{r}_0, \mathbf{t})$ is used in Bi-CGSTAB (10) for the calculation of ρ_{i+1} then a possible restart criterion is,

$$\frac{|(\mathbf{r}_0, \mathbf{t})|}{\|\mathbf{r}_0\|_2 \|\mathbf{t}\|_2} \leq (tol1). \tag{19}$$

A Joubert inner product test can also be used as a possible restart criterion for the sensitive value $(\mathbf{r}_0, \mathbf{v})$, i.e.

$$\frac{|(\mathbf{r}_0, \mathbf{v})|}{\|\mathbf{r}_0\|_2 \|\mathbf{v}\|_2} \leq (tol2). \tag{20}$$

By experiment, the monitors in (19) and (20) are found to become very small near the on-set of divergence. This indicates that these monitors can be used to signal when to restart in these problems.

Table 7 shows the harsh convergence performance with (19) and (20) used as restart criteria and $tol1 = tol2 = 10^5 \epsilon^{\frac{1}{2}}$, convergence is achieved in all cases.

Pe	ν	$(\|\mathbf{r}\|_2/\|\mathbf{f}\|_2)_{min}$	Iteration
	0.5	$< \epsilon$	8
	1	$< \epsilon$	24
	2	$< \epsilon$	24
1	5	$< \epsilon$	45
	10	$< \epsilon$	79
	20	$< \epsilon$	151
	40	$< \epsilon$	211
	0.5	$< \epsilon$	11
	1	$< \epsilon$	18
	2	$< \epsilon$	20
5	5	$< \epsilon$	46
	10	$< \epsilon$	74
	20	$< \epsilon$	112
	40	$< \epsilon$	147

Table 7: Performance with inner product criteria based restarts

Comparing the number of iterations required for convergence in Tables 6 and 7, the restart strategy based on (19) and (20) is more effective over the range of cases than a strategy based on any of the fixed restart periods. This is not surprising since the inner product based restart is effectively an adaptive strategy - only restarting when it senses the need for such action - while the fixed period strategy restarts regardless of whether it is required or not.

Figures 5 and 6 correspond to Figures 1 and 2 with the restarts based on (19) and (20). In these convergence histories, ○ denotes restart due to (19) and □ denotes restart due to (20).

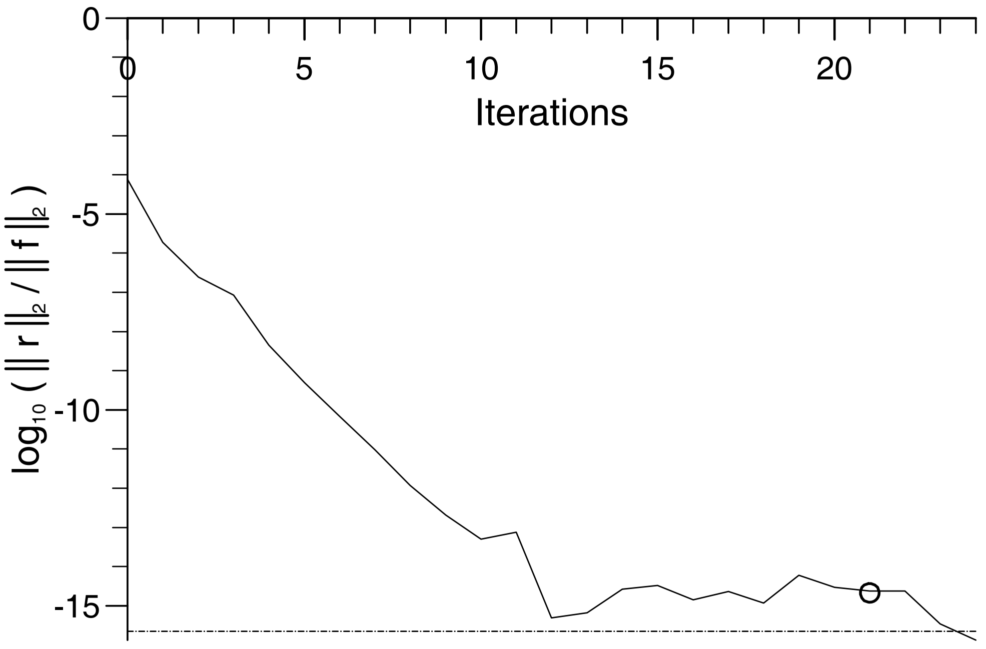

Figure 5: Inner product based restart for $\nu = 1$, $Pe = 1$

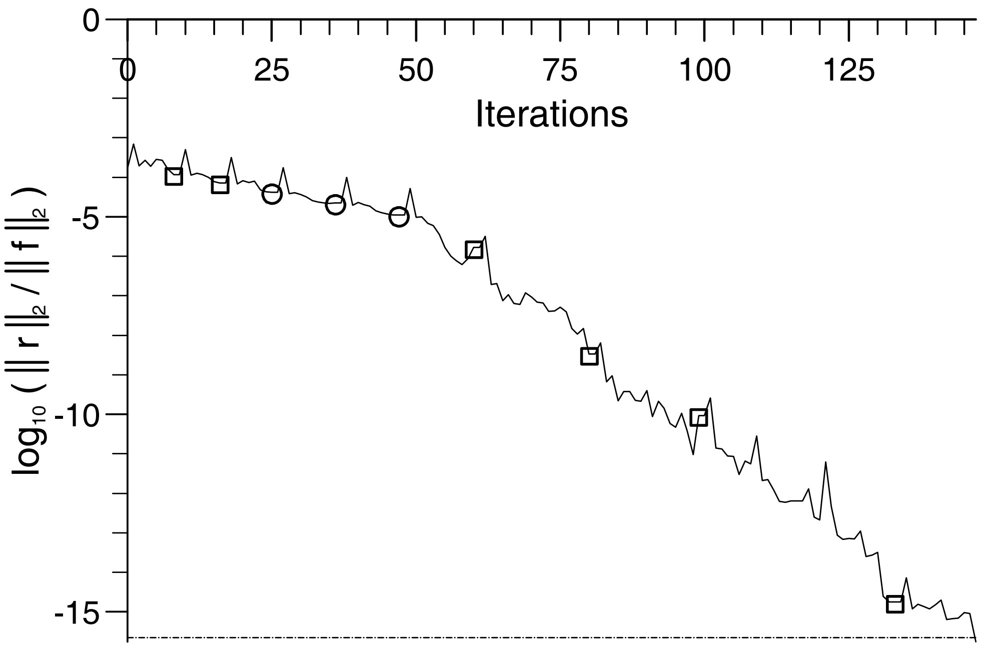

Figure 6: Inner product based restart for $\nu = 40$, $Pe = 5$

As with the requirement for the fixed restart period, k, in Section 5.3.1, the requirement for the parameters $tol1$ and $tol2$ violates the desired property of no external parameters.

6 Concluding Remarks

The use of implicit discretisation methods is essential for the long term transient solution of advection-diffusion equations. Due to the advection term, most implicit methods lead to linear systems which are nonsymmetric. A possible solver for these systems is Bi-CGSTAB. This solver has the attractive CG-like properties of efficiency, low storage and no external parameters. However, unlike CG, it has no minimisation property.

When examined by numerical experiment, the Bi-CGSTAB solver is as efficient as GMRES (which does possess a minimisation property) for the test case used in Section 4.1. However, the harsh convergence tests in Section 4.2 indicate that the robustness of Bi-CGSTAB is suspect under more extreme conditions. In these tests, rounding errors appear to corrupt the underlying recursion process and lead to divergence.

Of the methods used in Section 5 to improve the robustness of Bi-CGSTAB for the harsh convergence tests, neither residual smoothing nor random initial iterates attempt to control rounding error - these methods have only limited success in improving the convergence history.

The other method investigated for improving robustness is restarting. This method discards the Krylov subspace (based on some pre-determined criterion) and starts the iteration again from the latest iterate. At each restart, the rounding errors are removed from the process so their build-up is controlled - this method overcomes the divergence difficulties and the harsh convergence criterion is met in all the cases in Tables 6 and 7.

Various restart criteria are investigated in Section 5.3:

- Restarting at a fixed regular interval is effective but the rate of convergence is affected by the restart period (which must be supplied as an external parameter).
- Surprisingly, restart criteria based on the round-off unit definition appear to be of no use in detecting the on-set of convergence difficulties. This is unfortunate as there is no requirement for external parameters in this approach.
- Monitoring sensitive values in the algorithm and restarting when these fall below some prescribed tolerance, being an adaptive restarting strategy, leads to fast convergence but, in the tests carried out, the tolerance appears to have no physical meaning and so must be considered an external parameter.

Overall, restarting is an effective way of improving the robustness of the Bi-CGSTAB method. However, the requirement for an external parameter in the restart criterion is an unattractive feature of the method - it may be possible to overcome this with a better normalisation of the monitor value.

7 Acknowledgements

The author is financially supported by a NERC CASE studentship, the collaborating body being the Institute of Hydrology, Wallingford.

He wishes to thank Dr. M.J. Baines, Dr. N.K.Nichols and Dr. A.Priestley of the Mathematics Department, University of Reading, for their advice and assistance during this work.

References

[1] G. Brussino and V. Sonnad. “A comparison of direct and iterative techniques for sparse, unsymmetric systems of linear equations”. *Int. J. Numer. Meth. Engng.*, **28**:801–815, 1989.

[2] F.F. Campos,filho and J.S. Rollet. “Study of convergence behaviour of conjugate gradient-type methods for solving unsymmetric systems”. OUCL Report 92/24, Oxford University Computing Laboratory, 1992.

[3] A.J. Davies. *The Finite Element Method : A First Approach.* Clarendon Press, Oxford, 1980.

[4] V. Faber and T. Manteuffel. “Necessary and sufficient conditions for the existence of a conjugate gradient method”. *SIAM J. Numer. Anal.*, **21**(2):352–362, April 1984.

[5] R. Fletcher. “Conjugate gradient methods for indefinite systems”. *Lecture Notes in Math.*, **506**:73–89, 1976.

[6] R.W. Freund, G.H. Golub, and N.M. Nachtigal. “Iterative solution of linear systems”. *Acta Numerica*, pages 57–100, 1991.

[7] R.W. Freund and N.M. Nachtigal. “QMR : A quasi-minimal residual method for non-hermitian linear systems”. *Numer. Math.*, **60**:315–339, 1991.

[8] G.H. Golub and C.F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 1st edition, 1983.

[9] M.H. Gutknecht. “Variants of BiCGSTAB for matrices with complex spectrum”. *SIAM J. Sci. Stat. Comput.*, **14**(14), Sept. 1993. *To appear.*

[10] W. Joubert. *Generalized Conjugate Gradient and Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations.* PhD thesis, Center for Numerical Anaylsis, University of Texas, Austin, TX, January 1990.

[11] W. Joubert. “Lanczos methods fro the solution of nonsymmetric systems of linear equations”. *SIAM J. Matrix Anal. Appl.*, **13**(3):926–943, July 1992.

[12] C. Lanczos. “Solution of systems of linear equations by minimized iterations”. *J. Res. Natl. Bur. Stand.*, **45**:255–282, 1952.

[13] H.M. Leismann and E.O. Frind. “A symmetric-matrix time integration scheme for the efficient solution of advection-dispersion problems”. *Water Res. Res.*, **25**(6):1133–1139, 1989.

[14] N.M. Nachtigal, S.C. Reddy, and L.N. Trefethen. “How fast are nonsymmetric matrix iterations ?”. *SIAM J. Matrix Anal. Appl.*, **13**(3):778–795, July 1992.

[15] K.J. Neylon. “Application of the Taylor-Galerkin method to transport problems in subsurface hydrology”. Numerical Analysis Report 5/93, Maths. Dept., Univ. of Reading, 1993.

[16] B.N. Parlett, D.R. Taylor, and Z.A. Liu. "A look-ahead Lanczos algorithm for unsymmetric matrices". *Math. Comp.*, **44**(169):105–124, January 1985.

[17] A. Peters. "CG-like algorithms for linear systems stemming from the FE discretization of the advection-dispersion equation". In T.F. Russell, R.E. Ewing, C.A. Brebbia, W.G. Gray, and G.F. Pinder, editors, *Numerical Methods in Water Resources,* Proc. 9th Int. Conf. on Comp. Meth. in Water Res., University of Colorado, Denver, USA, pages 511–518, June 1992.

[18] G. Radicati, Y. Robert, and S. Succi. "Iterative algorithms for the solution of nonsymmetric systems in the modelling of weak plasma turbulence". *JCP*, **80**:489–497, 1989.

[19] Y. Saad and M.H. Schultz. "GMRES : A generalized minimum residual algorithm for solving non-symmetric linear systems". *SIAM J. Sci. Stat. Comput.*, **7**(3):856–869, 1986.

[20] G.L.G. Sleijpen and D.R. Fokkema. "BiCGSTAB(ℓ) for linear equations involving unsymmetric matrices with complex spectrum". Preprint NR 772, Department of Mathematics, University of Utrecht, March 1993.

[21] P. Sonneveld. "CG-S : A fast Lanczos-type solver for non-symmetric linear systems". *SIAM J. Sci. Stat. Comput.*, **10**(1):36–52, 1989.

[22] D.R. Taylor. *Analysis of the look ahead Lanczos algorithm.* PhD thesis, Department of Mathematics, University of California, Berkeley, CA, November 1982.

[23] H.A. van der Vorst. "BI-CGSTAB : A fast and smoothly convergent variant of BI-CG for the solution of nonsymmetric linear systems". *SIAM J. Sci. Stat. Comp.*, **13**(2):631–644, 1992.

[24] R. Weiss and W. Schönauer. "Accelerating generalized conjugate gradient methods by smoothing". In *Proceedings of IMACS '91, Brussels*, 1991. To appear.

[25] L. Zhou and H.F. Walker. "Residual smoothing techniques for iterative methods". Research report, Dept. of Maths. and Stats., Utah State University, December 1992.