Preconditioned iterative methods for implicit equations

A. J. Wathen
Oxford University Computing Laboratory

These notes are about methods for finding the solution x of the linear simultaneous equations

$$Ax = b \tag{1}$$

where A is a known real or complex $n \times n$ matrix (which is assumed invertible) and b is a known n-vector.

The coefficient matrix would generally be considered to be either full, banded or sparse. For any of these types of matrices, good numerical algorithms based upon systematic (Gaussian) elimination exist. (For general software, you need look no further than the NAG library. Hardware-specific routines exist on many advanced architecture machines.) Such algorithms terminate with (in general) an accurate numerical answer after a predictable ammount of computation. The basic difficulty is that for large n such predictable requirements in terms of CPU time, computer memory and disc storage may be completely unavailable whether you work on a workstation or a supercomputer. (See for example the table of work estimates on page 388 of [2]). However, iterative solution methods are often available and practicable for such large problems especially when the coefficient matrix is sparse.

In the context of dynamic simulations such as arise in weather prediction (as well as other areas including petroleum reservoir modelling, semiconductor process modelling and financial option pricing) such systems usually derive from an implicit time discretisation method, the simplest of which would be the backwards Euler method. Linearisation via Newtons method or Picard iteration might also be involved. In such implicit time-dependent computations the coefficient matrix always has the form

$$\frac{1}{\Delta t} I + \mathcal{L} \qquad \text{or} \qquad I + \Delta t \mathcal{L}$$

in the case of finite difference replacement of the spatial terms or

$$\frac{1}{\Delta t}\hat{M} + \mathcal{L} \qquad \text{or} \qquad \hat{M} + \Delta t\mathcal{L}$$

where $\hat{M}$ is the 'mass matrix' in the case of finite element approximation in the spatial variables - the appearance of $\hat{M}$ rather than I is of no great significance since it is known that simple diagonal scaling of $\hat{M}$ renders the solution of systems involving $\hat{M}$ trivial since $\text{diag}(\hat{\text{M}})^{-1}\hat{\text{M}}$ is (spectrally) very close to the identity (see [23]). For approximations based on expansions in terms of globally-defined functions (such as arise in spectral methods), the orthogonality properties of the basis functions are important. Here $\mathcal{L}$ represents the spatial differential opertor terms.

This is a simple truth, but from the point of view of solving the matrix system at each time step it indicates that the choice of time-step Δt can have a profound effect: if Δt is small then the coefficient matrix looks more and more like the identity matrix whereas larger values of Δt make the coefficient matrix closer to a steady-state operator. For direct (elimination) algorithms the choice of Δt does not affect the work in solving such systems (only the accuracy of the computed results) however for iterative methods, solution with small Δt generally takes very few iterations; many iterations might be required in the case of a larger time-step. Of course the choice of timestep is usually determined by the time scale of a simulation and any accuracy/stability constraints.

The two leading classes of iterative methods are those based on the Multigrid approach, and the Conjugate Gradient method and its generalisations.

The multigrid method was originally derived for grid-based problems such as arise from finite difference or finite element discretisation of partial differential equations. For certain model problems extremely rapid convergence rates are proved and observed with this method. For problems with no obvious geometric sequence of (multi-)grids, there has been attention directed towards finding a mimicking algebraic procedure (Algebraic Multigrid), but such methods are generally less rapidly convergent. Much research on Multigrid methods continues - I am not an expert on it! A general reference for Multigrid methods is [14].

Methods of Conjugate Gradient (CG) type (so-called Krylov subspace methods) are however more widely applicable, and convergence rates in many cases are fast. It is methods of this type that I mainly intend to review in these notes.

In all the methods of this type, reference to the coefficient matrix, A is only required for the computation of matrix×vector products, Ap (and sometimes also A^Tp) where p is a known n-vector. For a sparse matrix these products can usually be encoded so that only calculation on the non-zeros is required.

In many applications, *preconditioning* of the original linear system is employed. Indeed after an explosion of new Krylov subspace methods in the early 1990's it was realised that finding and effective preconditioner is much more important in general than the specific choice of iterative method. The idea here is that if a preconditioning matrix M can be easily computed such that convergence of the appropriate iterative method is faster for the preconditioned system

$$M^{-1}Ax = M^{-1}b \tag{2}$$

than the original, AND systems of equations with M as coefficient matrix are very rapidly solvable (such a solution will be required at *each* iteration), then the overall solution time will be shorter. The inverse of M is not required; the equation (2) is only schematically equivalent to the preconditioned methods that are used. Indeed a good preconditioner might be a 'process' such as a single multigrid cycle which only for theoretical purposes can be considered as a matrix.

A simple but not necessary motivation for a good preconditioner is that it be an easily invertible approximation of the coefficient matrix A. Alternatively one may have an approximate inverse so that preconditioning involves only a matrix×vector multiplication at each iteration.

There are many approaches to the construction of preconditioners. For particular applications problems, there is sometimes a specific and inherently simpler approximation which 'captures the main physics' and is more readily invertible which may be employed as a preconditioner. For example one might have computed a triangular factorisation but need solution of many linear systems with slightly different coefficient matrices - the original factorisation may then be used as a good preconditioner. This type of situation might arise for example in nonlinear/time-dependent problems.

For problems with particular mathematical structure, there are often ways to utilise that structure for preconditioning - for example for problems with constraints there have been some very successful recent advances which are based on approximation of subproblems (see [15],[12]).

For self-adjoint and positive definite elliptic partial differential equation

problems, successful preconditioners based on domain decomposition are also conveniently parallelisable. There is much work on these methods - see the sequence of proceedings of which the latest is can be obtained from http://www.gre.ac.uk/ dd11/dd11/proceedings.html.

Popular and general preconditioners for sparse systems can be constructed using incomplete triangular factorisation of A ([13],[9]) or by (parallel) construction of sparse approximate inverses ([3],[10]). The quality of these preconditioners varies when employed for different application problems and it can be expensive (even prohibitive) to construct them for very large scale problems. Less generally applicable preconditioners have also been suggested which work well on certain classes of problems (see [25],[11],[21]) or the special issue of Numerical Linear Algebra and Applications associated with the conference on Preconditioned held last summer in Minneapolis (see [16]).

We now turn to iterative methods.

For real symmetric (or complex hermitian) and positive definite systems, the (Hestenes-Steifel) Conjugate Gradient method ([7], page 370) is to set initial vectors $x_0 = 0, p_0 = 0$, and for each iteration

$$\begin{aligned} \text{compute residual} \quad & r_{k-1} = b - Ax_{k-1} \\ \text{compute conjugate search direction} \quad & p_k = r_{k-1} + \beta_k p_{k-1} \\ & \text{with } \beta_k = \frac{r_{k-1}^T r_{k-1}}{r_{k-2}^T r_{k-2}} \\ \text{update solution} \quad & x_k = x_{k-1} + \alpha_k p_k \\ & \text{with } \alpha_k = \frac{r_{k-1}^T r_{k-1}}{p_k^T A p_k}. \end{aligned}$$

The residual is usually computed recursively as

$$r_{k-1} = r_{k-2} - \alpha_{k-1} A p_{k-1}$$

so that only one matrix×vector multiplication is required per iteration.

The most readily obtained convergence bound for this method is that the exact solution x and the CG iterates x_k satisfy

$$\|x_k - x\|_A \leq 2 \left(\frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}} \right)^k \|x_0 - x\|_A \tag{3}$$

where κ is the spectral condition number of A (the ratio of largest to smallest eigenvalues of A), and $\|y\|_A^2 = y^T A y$. Thus rapid convergence is guaranteed if

κ is small. This is the motivation for preconditioning: to reduce κ. If a better guess for x_0 is available (for example the solution at a previous time-step), then this can be used and (3) indicates that few iterations are expected to refine the solution. Contrast this with direct solution procedures which take the same time whatever is known about the solution.

For real symmetric (or complex hermitian) *in*definite matrices and other matrices whose eigenvalues all lie on a single line in the complex plane, there exist Conjugate Gradient-like methods (for example the method of Conjugate Residuals of which probably the best of the various implementations is MINRES [17]) with similar convergence properties but which require a little more work per iteration. See the 'taxonomy' [1]. Similarly to Conjugate Gradients, rapid convergence can be guaranteed for problems with clusted eigenvalues - the precise theoretical convergence bounds/estimates are more complicated, see [8],[24].

For non-symmetric matrices, there are a number of competitive CG-like methods. None has the monotonic error- or residual-reduction *and* constant work per iteration like in the symmetric case, and convergence can be more erratic. Available convergence bounds are generally not descriptive in contrast to symmetric problems. Numerical divergence is occasionally seen. Nevertheless, a number of methods of this type are significantly used in a variety of situations.

The most longstanding of the non-symmetric methods is that based on (Hestenes-Steifel) CG for the normal equations

$$A^T A x = A^T b,$$

the coefficient matrix of which is symmetric and positive definite. This approach is not recommended unless $\kappa(A)$ is small as $\kappa(A^T A) = \kappa(A)^2$ and slow convergence as suggested by (3) is usually experienced. There are situations (such as implicit computations with a very small time-step or where a high-quality preconditioner is available) where this method might be used.

The GMRES method (Generalised Minimal Residual method) ([19]) theoretically is error-minimising at each iteration, however a Gram-Schmidt-like (in fact Arnoldi) orthogonalisation of the k^{th} 'search direction' p_k against all previous search direction vectors $p_0, p_1, \ldots, p_{k-1}$ is required at the k^{th} iteration - thus the work per iteration increases as the number of iterations grows. This is generally impractical in both storage as well as computation. The method may however be used in a cyclic manner: do l steps of GMRES

(storing and operating on l vectors) and then restart. This method is called GMRES(l). A popular value for l is about 10. The advantage of this method is that convergence can always be guaranteed by increasing l. The disadvantage is that too large a value for l may be required! Indeed, recent examples provided by Embree ([4]) indicate that it is quite possible for GMRES(l_1) to require fewer iterations and less work to obtain a solution to any convergence tolerance than GMRES(l_2) when $l_1 < l_2$. Thus increasing the restart length may not be profitable unless it is possible to employ the non-restarted method GMRES(n).

Restarting is generally not favoured if it can be helped, and in applications such as petroleum reservoir modelling where this method is widely employed (usually an implementation called Orthomin), the time-step is often kept small enough that few enough iterations are required so that restating is not necessary. In computations I have been involved with, it is not clear that the adaptive time-step selection mechanism is not in fact responding to the convergence rate of the iterative linear solver rather than any discretisation errors or non-convergence of the (outer) non-linear iteration.

The BiCG method ([5]) requires only 2 or 3 p-vectors, but computation of both of the products Ap and A^Tp is needed. (The latter may be awkward to compute depending on the data structure for storing A). The method can also stall and has been generally cast aside in favour of the modified methods below.

The CGS (CG Squared) method ([20]) is a variant of BiCG which avoids the calculation of the transpose product A^Tp. It exhibits rapid *final* convergence, though large jumps (increases) in the residual are usually seen at earlier stages of the iteration.

A further variant of BiCG is the BiCG-STAB method ([22]) which is similar to CGS, but has a 'more monotonic' residual reduction. A useful hybrid combination of a few steps of GMRES with BiCG-STAB leads to the useful BiCG-STAB(l) method ([18]). Both CGS and BiCG-STAB and its variants are widely used in semiconductor device simulations.

The QMR (Quasi Minimal Residual) method ([6]) is related to both BiCG and GMRES looked a very attractive approach when it was announced but despite the availability of good software, it seems not to be commonly used except for certain model reduction problems in circuit simulation.

The GMRES(l), CGS and BiCG-STAB(l) methods with appropriate preconditioning are all significantly used in a number of practical situations not indicated here.

References

[1] S.F. ASHBY, T.A. MANTEUFFEL AND P.E. SAYLOR, *A taxonomy for conjugate gradient methods,* SIAM J. Numer. Anal. **27** (1990), pp. 1542-1568.

[2] O. AXELSSON AND V.A. BARKER, *Finite Element Solution of Boundary Value Problems: Theory and Computation,* Academic Press, New York, 1984.

[3] M. BENZI AND M. TUMA, *A sparse approximate inverse preconditioner for nonsymmetric linear systems,* SIAM J. Sci. Comput. **19** (1998), pp. 968-994.

[4] M. EMBREE, *Convergence of Krylov subspace methods for non-normal matrices* D. Phil thesis, Oxford University, 1999.

[5] R. FLETCHER, *Conjugate Gradient Methods for Indefinite Systems,* in Numerical Analysis, Dundee 1975, G.A Watson, ed. Springer, New York, 1976.

[6] R.W. FREUND AND N.M. NACHTIGAL, *QMR: a quasi-minimal residual method for non-hermitian linear systems,* Numer. Math. **60** (1991), pp. 315-339.

[7] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations,* John Hopkins University Press, Baltimore, 1983.

[8] A. GREENBAUM, *Iterative methods for solving linear systems,* SIAM, Philadelphia, 1997.

[9] I. GUSTAFSSON, *A Class of First Order Factorization Methods,* BIT **18** (1978), pp. 142-156.

[10] M. J. GROTE AND T. HUCKLE, *Parallel preconditioning with sparse approximate inverses,* SIAM J. Sci. Comput. **18** (1997), pp. 838-853.

[11] T.J.R HUGHES, I. LEVIT AND J. WINGET, *An Element-by-Element Solution Algorithm for Problems of Structural and Solid Mechanics,* Comput. Meths. Appl. Mech. Engrg **36** (1983), pp. 241-254.

[12] C. KELLER, N.I.M. GOULD, AND A.J. WATHEN, *Constraint preconditioning for indefinite linear systems,* SIAM J. Matrix Anal. Appl. **21**(4) (2000), pp. 1300-1317.

[13] J.A. MEIJERINK AND H.A. VAN DER VORST, *An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix,* Math. Comput. **31** (1977), pp. 148-162.

[14] S.F. MCCORMICK, *Multigrid Methods,* SIAM, Philadelphia, 1987.

[15] M.F. MURPHY, G.H. GOLUB AND A.J. WATHEN, 2000, *A note on preconditioning for indefinite linear systems,* SIAM J. Sci. Comput., to appear.

[16] Numer. Linear Algebra Appl. (Autumn 2000 issue).

[17] C.C. PAIGE AND M.A. SAUNDERS, *Solution of Sparse Indefinite Systems of Linear Equations,* SIAM J. Numer. Anal. **12** (1975), pp. 617-629.

[18] G.L.G. SLEIJPEN AND D.R. FOKKEMA, *BiCGstab(L) for linear equations involving unsymmetric matrices with complex spectrum,* Electronic Trans. Numeri. Anal.,**1** (1993), pp. 11-32.

[19] Y. SAAD AND M.H. SCHULTZ, *GMRES: A generalised minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Stat. Comput. **7** (1986), pp. 856-869.

[20] P. SONNEVELD, *CGS: a fast Lanczos-type solver for non-symmetric linear systems,* SIAM J. Sci. Stat. Comput. **10** (1989), pp 36-52.

[21] G. STRANG, *A proposal for Toeplitz matrix calculation,* Studies Appl. Math. **74** (1986), pp. 171-176.

[22] H.A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly convergent variant of BiCG for the solution of non-symmetric linear systems,* SIAM J. Sci. Stat. Comput. **13** (1992), pp. 631-644.

[23] A.J. WATHEN, *Realistic eigenvalue bounds for the Galerkin Mass Matrix,* IMA J. Numer. Anal. **7** (1987), pp. 449-457.

[24] A.J. WATHEN, B. FISCHER AND D.J. SILVESTER, *The convergence of iterative solution methods for symmetric and indefinite linear systems,* in 'Numerical Analysis 1997', Eds. D.F. Griffiths & G.A. Watson, Pitman Research Notes in Mathematics Series, Addison Wesley Longman, Harlow, England, 1998, pp. 230-243.

[25] H. YSERENTANT, *On the Multi-level Splitting of Finite Element Spaces,* Numer. Math. **49** (1986), pp. 379-412.