Preconditioning of Symmetric, but Highly Indefinite Linear Systems[1]

Roland W. Freund
Bell Laboratories
Room 2C-420, 700 Mountain Avenue
Murray Hill, NJ 07974–0636, USA
e-mail: freund@research.bell-labs.com

Keywords: linear system, symmetric QMR iteration, KKT system, SSOR, Stokes problem

ABSTRACT

Many important applications involve the solution of large linear systems with symmetric, but highly indefinite coefficient matrices with both many positive and many negative eigenvalues. For example, such systems arise in incompressible flow computations and as subproblems in optimization algorithms for linear and nonlinear programs. Often, these systems are very large and sparse, and it then becomes attractive to use a preconditioned iterative method, such as the symmetric QMR algorithm, for their solution. In this paper, we first briefly review the symmetric QMR algorithm, and we then discuss the construction of efficient preconditioners for highly indefinite symmetric linear systems. We report a few results of numerical experiments with symmetric indefinite linear systems arising in linear programming and in fluid-flow computations.

INTRODUCTION

There are important applications that require the solution of large sparse systems of linear equations

$$\mathbf{K}\,\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{K} = \mathbf{K}^{\mathrm{T}} \in \mathbb{R}^{N\times N}$ is symmetric, but *highly indefinite*, i.e., $\mathbf{K}$ has both *many positive* and *many negative* eigenvalues.
As a first example, consider the Stokes problem of determining the velocity field $\mathbf{u}$ and pressure p of an incompressible fluid in a flow region $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. The flow is described

[1]Numerical Analysis Manuscript No. 97–3–03, Bell Laboratories, Murray Hill, New Jersey, May 1997. Available on WWW at {`http://cm.bell-labs.com/cs/doc/97`}.

by the coupled partial differential equations

$$\begin{aligned} -\nu\nabla^2\mathbf{u} + \operatorname{grad} p &= \mathbf{f} \quad \text{in } \Omega, \\ \operatorname{div} \mathbf{u} &= 0 \quad \text{in } \Omega, \\ &+ \quad \text{appropriate boundary conditions}, \end{aligned} \tag{2}$$

where $\nu > 0$ denotes the viscosity of the fluid. Finite-difference or finite-element approximations of (2) lead to linear systems (1) with matrices $\mathbf{K}$ of the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{B} & \mathbf{C}^{\mathrm{T}} \\ \mathbf{C} & -\mathbf{D} \end{bmatrix}, \quad \mathbf{B} = \mathbf{B}^{\mathrm{T}} \in \mathbb{R}^{n\times n}, \quad \mathbf{D} = \mathbf{D}^{\mathrm{T}} \in \mathbb{R}^{(N-n)\times(N-n)}; \tag{3}$$

see, e.g., [8, 11]. The matrix $\mathbf{B}$ in (3) represents the discretization of the term $-\nu\nabla^2\mathbf{u}$ in (2), and thus $\mathbf{B}$ is positive definite. Furthermore, the matrix $\mathbf{C}$ in (3) usually has full row rank, and $\mathbf{D}$ is a positive semi-definite matrix. These properties imply that $\mathbf{K}$ is an indefinite matrix with n positive and $N-n$ negative eigenvalues, and since $n \gg 1$ and $N-n \gg 1$ the matrix $\mathbf{K}$ is highly indefinite.

A second application is interior-point methods for linear programming. Consider a linear program (LP) in standard form:

$$\min_{\mathbf{u}\in\mathbb{R}^n} \mathbf{c}^{\mathrm{T}}\mathbf{u} \qquad \text{subject to} \qquad \mathbf{A}\,\mathbf{u} = \mathbf{d} \quad \text{and} \quad \mathbf{u} \geq 0. \tag{4}$$

Here, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{d} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$ are given. A vector $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ is a solution of (4) if, and only if, there exist $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \cdots & s_n \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ such that

$$\mathbf{A}\,\mathbf{u} = \mathbf{d}, \quad \mathbf{A}^{\mathrm{T}}\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad \text{and} \quad u_i \geq 0, \quad s_i \geq 0, \quad u_i s_i = 0 \quad \text{for all} \quad i = 1, 2, \ldots, n. \tag{5}$$

Essentially, interior-point methods for solving the LP (4) can be viewed as variants of Newton's method applied to the system of equations (5) with u_i, $s_i \geq 0$ replaced by strict positivity u_i, $s_i > 0$; see, e.g., [3] and the references given there. Each Newton iteration requires the solution of a linear system (1) with a matrix $\mathbf{K}$ of the form

$$\mathbf{K} = \begin{bmatrix} -\mathbf{D} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{bmatrix}, \quad \text{where} \quad \mathbf{D} = \operatorname{diag}\left(\frac{s_1}{u_1}, \frac{s_2}{u_2}, \ldots, \frac{s_n}{u_n}\right). \tag{6}$$

The matrix $\mathbf{D}$ in (6) is positive definite, and it changes with each interior-point iteration. The matrix $\mathbf{A}$ in (6) is fixed. Assuming that $\operatorname{rank}\mathbf{A} = m$, the matrix $\mathbf{K}$ has exactly m positive and n negative eigenvalues.

Symmetric indefinite linear systems (1) also arise in the solution of nonlinear programs. There, $\mathbf{K}$ is of the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{H} & \mathbf{G}^{\mathrm{T}} \\ \mathbf{G} & \mathbf{0} \end{bmatrix}, \quad \mathbf{H} = \mathbf{H}^{\mathrm{T}} \in \mathbb{R}^{n\times n},$$

where, in general, both $\mathbf{H}$ and $\mathbf{G}$ change with each iteration of the optimization algorithm.

THE SYMMETRIC QMR ALGORITHM

When iterative methods are used to solve linear systems, it is crucial to combine them with efficient preconditioning. Let

$$\mathbf{M} = \mathbf{M}^{\mathrm{T}} = \mathbf{M}_1 \cdot \mathbf{M}_2 \in \mathbb{R}^{N \times N} \tag{7}$$

be a suitable preconditioner for (1), i.e., $\mathbf{M}$ should in some sense "approximate" $\mathbf{K}$, while linear systems with $\mathbf{M}_1$ and $\mathbf{M}_2$ are "easy" to solve compared to solving (1). Instead of applying an iterative method to the original system (1), the iteration is applied (at least implicitly) to the preconditioned system

$$\mathbf{M}_1^{-1}\mathbf{K}\,\mathbf{M}_2^{-1}\,(\mathbf{M}_2\,\mathbf{x}) = \mathbf{M}_1^{-1}\mathbf{d}.$$

Preconditioned conjugate gradient-type algorithms are powerful iterative schemes for the solution of large linear systems. For the case of symmetric indefinite systems, the standard algorithms of this type are SYMMLQ and MINRES [9]. However, both require that $\mathbf{M}_2 = \mathbf{M}_1^{\mathrm{T}}$ in (7), and thus they are restricted to symmetric positive definite preconditioners $\mathbf{M}$. For symmetric, but highly indefinite systems, this restriction on the preconditioner is rather unnatural, and instead, one would like to be able to choose a preconditioner (7) that is also highly indefinite in order to better approximate $\mathbf{K}$.
A conjugate gradient-type method for (1) that allows the use of an arbitrary symmetric nonsingular preconditioner (7) is the symmetric QMR algorithm [6]. This algorithm is an adaptation of the general nonsymmetric QMR method [5] to exploit the symmetry of the matrix $\mathbf{K}$ in (1). If the preconditioner $\mathbf{M}$ is indefinite, then the iterate of symmetric QMR are characterized by a quasi-minimization of the preconditioned residual norm. If $\mathbf{M}$ is positive definite, then the residual norms are truly minimized, and in fact, symmetric QMR and MINRES are mathematically equivalent in this case. Next, we present a statement of the symmetric QMR algorithm in its simplest form, namely without look-ahead.

Algorithm 1 (Symmetric QMR algorithm for solving the symmetric system $\mathbf{K}\,\mathbf{x} = \mathbf{b}$, using a symmetric preconditioner $\mathbf{M} = \mathbf{M}_1 \cdot \mathbf{M}_2$)

Start: *Choose* $\mathbf{x}_0 \in \mathbb{R}^N$ *and set*

$$\mathbf{s}_0 = \mathbf{b} - \mathbf{K}\,\mathbf{x}_0, \quad \mathbf{t} = \mathbf{M}_1^{-1}\mathbf{s}_0, \quad \mathbf{q}_0 = \mathbf{M}_2^{-1}\mathbf{t}, \quad \tau_0 = \|\mathbf{t}\|_2, \quad \vartheta_0 = 0, \quad \rho_0 = \mathbf{s}_0^{\mathrm{T}}\mathbf{q}_0.$$

For $j = 1, 2, \ldots,$ ***do***:

1) *Compute*

$$\mathbf{t} = \mathbf{K}\,\mathbf{q}_{j-1}, \quad \sigma_{j-1} = \mathbf{q}_{j-1}^{\mathrm{T}}\mathbf{t}, \quad \alpha_{j-1} = \frac{\rho_{j-1}}{\sigma_{j-1}}, \quad \mathbf{s}_j = \mathbf{s}_{j-1} - \alpha_{j-1}\,\mathbf{t}.$$

2) *Compute*

$$\mathbf{t} = \mathbf{M}_1^{-1}\mathbf{s}_j, \quad \vartheta_j = \frac{\|\mathbf{t}\|_2}{\tau_{j-1}}, \quad c_j = \frac{1}{\sqrt{1+\vartheta_j^2}}, \quad \tau_j = \tau_{j-1}\,\vartheta_j\,c_j,$$

$$\mathbf{d}_j = c_j^2\,\vartheta_{j-1}^2\,\mathbf{d}_{j-1} + c_j^2\,\alpha_{j-1}\,\mathbf{q}_{j-1}.$$

3) *Set*

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \mathbf{d}_j.$$

If $\mathbf{x}_j$ *has converged, then stop.*

4) *Compute*

$$\mathbf{u}_j = \mathbf{M}_2^{-1}\mathbf{t}, \quad \rho_j = \mathbf{s}_j^{\mathrm{T}}\mathbf{u}_j, \quad \beta_j = \frac{\rho_j}{\rho_{j-1}}, \quad \mathbf{q}_j = \mathbf{u}_j + \beta_j\,\mathbf{q}_{j-1}.$$

PRECONDITIONING

In this section, we briefly describe two strategies for constructing suitable symmetric indefinite preconditioners (7) for linear systems (1). The first preconditioner is a block variant of SSOR [1]. The second one is an adaptation of the ILUT preconditioner [10] for nonsymmetric systems to the symmetric case.

Block SSOR

Here, one first determines a permutation $\mathbf{\Pi}$ such that

$$\mathbf{\Pi}\,\mathbf{K}\,\mathbf{\Pi}^{\mathrm{T}} = \mathbf{L} + \mathbf{\Lambda} + \mathbf{L}^{\mathrm{T}}, \tag{8}$$

where $\mathbf{\Lambda}$ is nonsingular and "close" to diagonal, and $\mathbf{L}$ is strictly lower triangular. Given the decomposition (8), one then uses

$$\mathbf{M} = \mathbf{\Pi}^{\mathrm{T}}\left(\mathbf{\Lambda} + \mathbf{L}\right)\mathbf{\Lambda}^{-1}\left(\mathbf{\Lambda} + \mathbf{L}^{\mathrm{T}}\right)\mathbf{\Pi} \tag{9}$$

as a preconditioner for

$$\mathbf{K} = \mathbf{\Pi}^{\mathrm{T}}\left(\mathbf{L} + \mathbf{\Lambda} + \mathbf{L}^{\mathrm{T}}\right)\mathbf{\Pi}. \tag{10}$$

Subtracting (9) and (10), we obtain the "error" of the SSOR preconditioner:

$$\mathbf{M} - \mathbf{K} = \mathbf{\Pi}^{\mathrm{T}}\,\mathbf{L}\,\mathbf{\Lambda}^{-1}\mathbf{L}^{\mathrm{T}}\,\mathbf{\Pi},$$

which suggests to choose $\mathbf{\Pi}$ in (8) such that $\mathbf{\Lambda}$ is in some sense as "large" as possible. For indefinite matrices $\mathbf{K}$, the choice of $\mathbf{\Lambda}$ gets complicated by the fact that, in general, $\mathbf{\Lambda}$ cannot be chosen as a scalar diagonal matrix. For example, consider the matrices $\mathbf{K}$ in (6) that arise in interior-point methods for LP's. Assuming that $\mathbf{A}$ in (6) has rank m, it is easy to see that any potential matrix $\mathbf{\Lambda}$ in (8) has to be block diagonal with precisely m 2×2 blocks and $n-m$ 1×1 blocks. Strategies for determining a "near-optimal" $\mathbf{\Lambda}$ in this case are described in [3, 4].

Finally, we remark that a main advantage of block SSOR is that each multiplication $\mathbf{K} \cdot \mathbf{q}$ in the symmetric QMR algorithm can essentially be reduced to a multiplication $\mathbf{\Lambda} \cdot \mathbf{q}$ with the block-diagonal matrix $\mathbf{\Lambda}$ by using the so-called Eisenstat trick [2].

Incomplete $\mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}}$ factorization

Here, one computes an "incomplete" factorization

$$\mathbf{\Pi}\,\mathbf{K}\,\mathbf{\Pi}^{\mathrm{T}} = \mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}} + \mathbf{E}, \tag{11}$$

where $\mathbf{\Lambda}$ is block diagonal with 1×1 and 2×2 blocks and $\mathbf{L}$ is block lower triangular. In (11), $\mathbf{\Pi}$ is a permutation matrix whose rôle is twofold. First, if necessary, it is used to set up sufficiently nonsingular 2×2 blocks in $\mathbf{\Lambda}$, and second, it is used to pivot for sparsity to reduce potential fill-in in $\mathbf{\Lambda}$. Furthermore, the sparsity of $\mathbf{L}$ is controlled by "throwing away" unwanted elements, i.e., by putting them into the remainder matrix $\mathbf{E}$ in (11). Given the decomposition (11), one uses $\mathbf{M} = \mathbf{\Pi}^{\mathrm{T}}\,\mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}}\,\mathbf{\Pi}$ as a preconditioner for $\mathbf{K}$.

NUMERICAL EXAMPLES

In this section, we present two numerical examples. The first example is a linear system (1) with matrix (3) that arises from a 2-dimensional Stokes problem (2). We solved this system both with symmetric QMR and MINRES. For QMR, we used the (indefinite) diagonal part $\mathbf{M} = \mathrm{diag}(\mathbf{K})$ of (3) as the preconditioner. For MINRES, we used $\mathbf{M} = \mathbf{M}_+ = |\mathrm{diag}(\mathbf{K})|$ as the preconditioner. In Figure 1, we show the corresponding convergence curves. Since work and storage per iteration is roughly the same for QMR and MINRES, QMR is clearly superior.

The second example is the LP ISRAEL from the netlib test suite [7] of linear programs. Recall that at each interior-point iteration, a linear system (1) has to be solved. In Figure 2, we compare the number of QMR iterations for each of the linear systems solved in a complete run of the interior-point method, using block SSOR and the incomplete $\mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}}$ preconditioner (with various levels of fill-in). Note that block SSOR is quite efficient in the early stages of the interior-point method, but cannot compete with incomplete $\mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}}$ in the last few interior-point iterations. For the system at one of the interior-point iterations, we show the spectra of the preconditioned matrices in Figure 3. Clearly, incomplete $\mathbf{L}\,\mathbf{\Lambda}\,\mathbf{L}^{\mathrm{T}}$ clusters the spectrum much better than block SSOR.

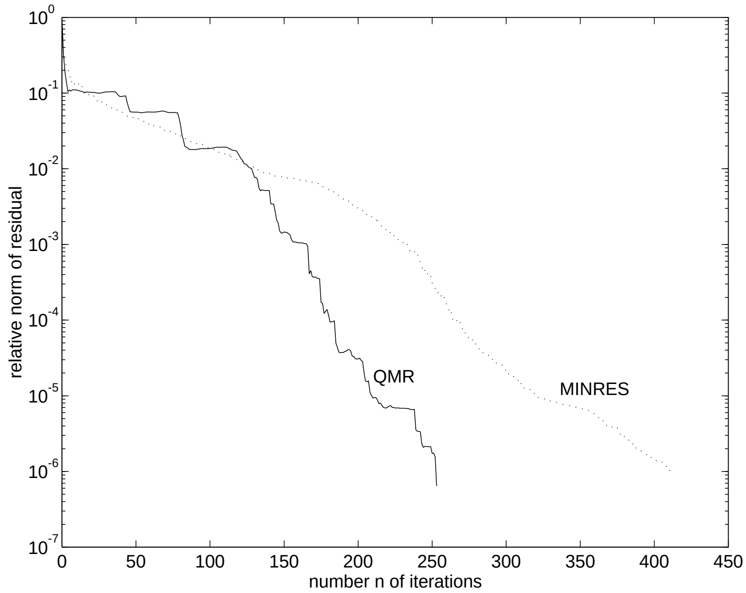

Figure 1: Convergence cureves for Stokes example.

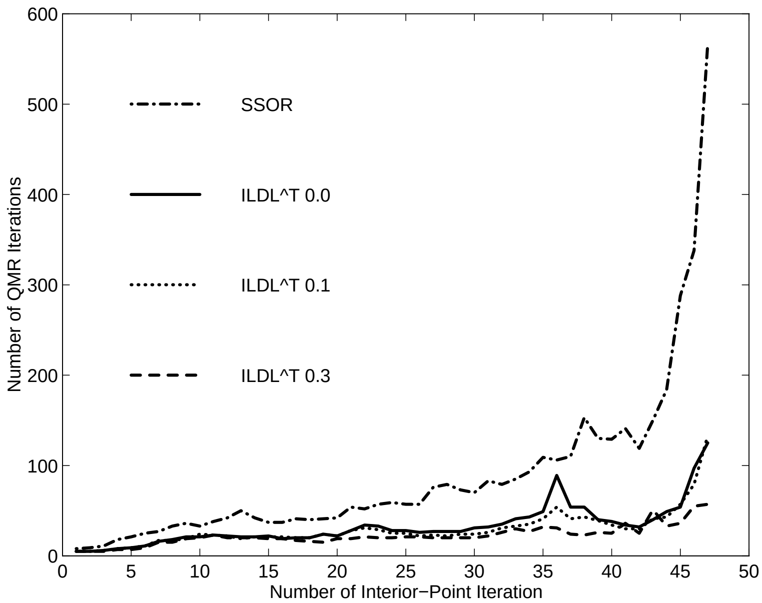

Figure 2: Interior-point runs with different preconditioners for LP ISRAEL.

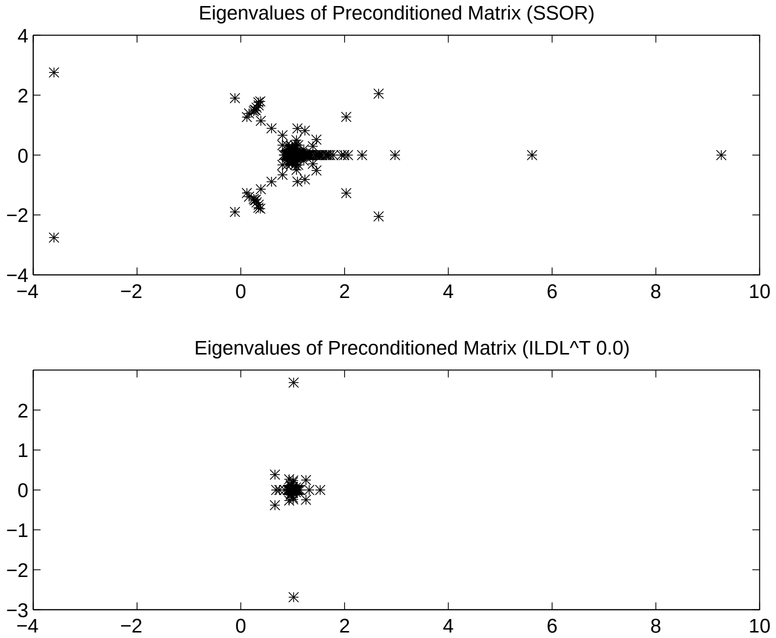

Figure 3: Spectra of preconditioned matrices for LP ISRAEL.

References

[1] O. Axelsson, *A survey of preconditioned iterative methods for linear systems of algebraic equations*, BIT 25, p.166 (1985).

[2] S.C. Eisenstat, *Efficient implementation of a class of preconditioned conjugate gradient methods*, SIAM J. Sci. Statist. Comput. 2, p.1 (1981).

[3] R.W. Freund, F. Jarre, *A QMR-based interior-point algorithm for solving linear programs*, Math. Programming 76, p.183 (1997).

[4] R.W. Freund, F. Jarre, *Numerical experiences with a QMR-based interior-point algorithm for linear programs*, in preparation.

[5] R.W. Freund, N.M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math. 60, p.315 (1991).

[6] R.W. Freund, N.M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math. 19, p.319 (1995).

[7] D.M. Gay, *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society COAL Newsletter 13 p.10 (1985).

[8] M.D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, San Diego (1989).

[9] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal. 12 p.617 (1975).

[10] Y. Saad, *ILUT: a dual threshold incomplete LU factorization*, Numerical Linear Algebra with Applications 1 p. 387 (1994).

[11] D.J. Silvester, K. Kechkar, *Stabilised bilinear-constant velocity-pressure finite elements for the conjugate gradient solution of the Stokes problem*, Comput. Methods Appl. Mech. Engrg. 79, p.71 (1990).