Acta Numerica (2015), pp. 329–376
doi:10.1017/S0962492915000021

Preconditioning

A. J. Wathen
Mathematical Institute, University of Oxford,
Andrew Wiles Building, Radcliffe Observatory Quarter,
Woodstock Road, Oxford OX2 6GG, UK
E-mail: wathen@maths.ox.ac.uk

The computational solution of problems can be restricted by the availability of solution methods for linear(ized) systems of equations. In conjunction with iterative methods, preconditioning is often the vital component in enabling the solution of such systems when the dimension is large. We attempt a broad review of preconditioning methods.

CONTENTS

1. Introduction

This review aims to give a comprehensive view of the current state of preconditioning for general linear(ized) systems of equations. However, we start with a specific example.

Consider a symmetric matrix with constant diagonals:

$$A = \begin{bmatrix} a_0 & a_1 & \cdot & \cdot & a_{n-1} \\ a_1 & a_0 & a_1 & \cdot & \cdot \\ \cdot & a_1 & a_0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & a_1 \\ a_{n-1} & \cdot & \cdot & a_1 & a_0 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Such a matrix is called a symmetric Toeplitz matrix. It is clearly defined by just n real numbers, but what is the best solution method for linear systems involving A? The fastest known stable factorization (direct) methods for solving a linear system $Ax = b$ for $x \in \mathbb{R}^n$ given $b \in \mathbb{R}^n$ require $\mathcal{O}(n^2)$ floating point computations. This is still better than the $\mathcal{O}(n^3)$ computations required for a general real $n \times n$ matrix using the well-known Gauss elimination algorithm.

Strang (1986) observed that one could approximate A by the circulant matrix

$$C = \begin{bmatrix} a_0 & a_1 & \cdots & a_{\lfloor \frac{n}{2} \rfloor} & a_{\lfloor \frac{n-1}{2} \rfloor} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_1 & \cdots & a_{\lfloor \frac{n}{2} \rfloor} & a_{\lfloor \frac{n-1}{2} \rfloor} & \cdots & a_2 \\ \cdots & a_1 & a_0 & \ddots & \cdots & a_{\lfloor \frac{n}{2} \rfloor} & \ddots & \cdots \\ a_{\lfloor \frac{n}{2} \rfloor} & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & a_{\lfloor \frac{n-1}{2} \rfloor} \\ a_{\lfloor \frac{n-1}{2} \rfloor} & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & a_{\lfloor \frac{n}{2} \rfloor} \\ \cdots & \ddots & a_{\lfloor \frac{n}{2} \rfloor} & \cdots & \ddots & a_0 & a_1 & \cdots \\ a_2 & \cdots & a_{\lfloor \frac{n-1}{2} \rfloor} & a_{\lfloor \frac{n}{2} \rfloor} & \cdots & a_1 & a_0 & a_1 \\ a_1 & a_2 & \cdots & a_{\lfloor \frac{n-1}{2} \rfloor} & a_{\lfloor \frac{n}{2} \rfloor} & \cdots & a_1 & a_0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where the outer diagonals of A have been overwritten by the central diagonals, which are 'wrapped around' in a periodic manner. Circulant matrices are diagonalized by a Fourier basis (Van Loan 1992) and it follows that for any given $b \in \mathbb{R}^n$, the fast Fourier transform (FFT) (Cooley and Tukey 1965) enables the computation of the matrix–vector product Cb and of the solution, $y \in \mathbb{R}^n$, of $Cy = b$ in $\mathcal{O}(n \log n)$ arithmetic operations. The importance of Strang's observation is that C is close to A if the discarded entries $a_{\lceil n/2 \rceil}, \dots, a_n$ are small; precise statements will describe this in terms of the decay in the entries of A along rows/columns moving away from the diagonal. With such a property, one can show that not only are the matrices A and C close entrywise, but that they are spectrally close, that is, the eigenvalues of $C^{-1}A$ (equivalently the eigenvalues λ of the matrix 'pencil' $A - \lambda C$) are mostly close to 1. As a consequence, an appropriate iterative method (it would be the conjugate gradient method if A is positive definite: see below) will compute a sequence of vectors $x_1, x_2, \ldots$ which converge rapidly from any starting guess, x_0, to the solution x of $Ax = b$. At each iteration, only a matrix–vector product with A and the solution of a single system with C, as well as vector operations, need to be computed. For any desired accuracy, a fixed number of iterations independent of the dimension n will be required. The outcome is that use of the FFT for solving systems with C and for doing multiplications of vectors by A (since A can in an obvious way be embedded in a circulant matrix of dimension $2n-1 \times 2n-1$) guarantees that the iterative method will require only $\mathcal{O}(n \log n)$ operations to compute the solution.

Strang's 'proposal for Toeplitz matrix calculations' is thus to employ the circulant matrix C as a *preconditioner* for linear systems involving A. One can naively think of the preconditioning as converting a linear system

$$Ax = b,$$

which is not so readily solved into the system

$$C^{-1}Ax = C^{-1}b$$

for which there is a fast solution method. This is not exactly what is done: generally $C^{-1}A$ will be non-symmetric even though A and C are symmetric, but it is possible to precondition with C whilst preserving symmetry of the preconditioned system matrix; see Saad (2003, Algorithm 9.1).

This example captures the essence of *preconditioning*. One does not always have matrices that are so obviously structured (nor with potentially so many non-zero entries), nor does one always have a clever and efficient tool such as the FFT, but, especially for matrices of large dimension, it can be extremely beneficial to have a good matrix approximation, that is, a good preconditioner, when solving linear systems of equations. Indeed, such a preconditioner is often necessary in order to enable practical computation of large-scale problems within a reasonable time on any given computational platform. The reduction from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$ in the above example brings some large-dimensional Toeplitz systems into the range of practical computation.

One does not necessarily need to have a preconditioner as a matrix: a linear operation (*i.e.*, a computational procedure which applies a linear operation to a vector) is what is required.

This article is a personal perspective on the subject of preconditioning. It addresses preconditioning only in the most common context of the solution of linear systems of equations. Much research on this problem has been directed at sparse matrices coming from the discretization of partial differential equations (PDEs) by finite element, finite volume or finite difference methods. This is certainly one major source for such problems, but it is far from the only area in which large-dimensional linear systems arise. It is true, however, that underlying properties of a PDE problem are often reflected in the matrices which arise from discretization, so that matrices arising from PDEs can inherit structural properties that allow them to be considered not simply as arrays of numbers. Indeed, there are many PDE analysts who consider PDE matrices to be simply members of families of discrete operators with many of the properties of the underlying continuous operators from which they are derived through discretization; see for example Mardal and Winther (2011), or the recent book by Málek and Strakoš (2014). This can be a very valuable viewpoint. Indeed, from this perspective, not applying any preconditioning is seen to coincide with a rather unnatural choice of mappings on the underlying spaces (Günnel, Herzog and Sachs 2014). Sometimes one comes across the perception that *multigrid* must be the most efficient method to solve any PDE problem. This is far from the truth, but multigrid methods and ideas remain very important both as solvers for certain simpler problems, and, in particular, as components of preconditioners for a wider class of problems, as we shall describe.

Whether or not a matrix system arises from PDEs, knowledge of the source problem can be helpful in designing a good preconditioner. Purely algebraic approaches which simply take the numerical matrix entries as input can have merit in situations where little is known about the underlying problem; they certainly have the useful property that it is usually possible to apply such a method! However, the generated preconditioners can be poor. This division between approaches to preconditioning essentially comes down to whether one is 'given a problem' or 'given a matrix' (Neytcheva 1999). Though we emphasize much about the former, we also give pointers to and a brief description of the latter.

Preconditioners are overwhelmingly used with Krylov subspace iterative methods (van der Vorst 2003, Greenbaum 1997, Saad 2003, Meurant 1999, Hackbusch 1994, Elman, Silvester and Wathen 2014*b*, Liesen and Strakoš 2013, Olshanskii and Tyrtyshnikov 2014), and the applicable iterative methods make different requirements of a preconditioner: for example, one could employ a non-symmetric preconditioner for a symmetric matrix, but in general one would then lose the possibility of using the very effective conjugate gradient or MINRES Krylov subspace iterative methods. Therefore, we have structured this paper into sections based on the symmetry and definiteness properties of the presented coefficient matrix. In essence, this subdivision can also be thought of in terms of the Krylov subspace method of choice.

The coefficient matrix will usually be denoted by A, as in the example above (though different fonts will be used). For any invertible preconditioning matrix or linear operator P, there is an iterative method that can be used, but it is usually desirable not to ignore structural properties such as symmetry when choosing/deriving a preconditioner.

We have generally restricted consideration to real square matrices, A, but it will be evident that some of the different preconditioning ideas could be applied to matrices over different algebraic fields, in particular the complex numbers (for which 'symmetric' → 'Hermitian', *etc.*). It is also true that some of the ideas presented apply to rectangular matrices, such as those arising in least-squares problems, but, except tangentially, this paper does not consider such situations.

Our general strategy/intention is to describe most preconditioning approaches that we are aware of, and to guide the reader towards selecting an appropriate strategy for any particular problem. Inevitably, some preconditioning ideas have arisen in the specific context of particular problems, so a brief review of such problems is included.

More than ten years ago, Benzi (2002) wrote an excellent survey of preconditioning techniques, which contains much wisdom and remains an excellent source today, in particular with regard to incomplete factorization and sparse approximate inverse methods. The book by Chen (2005) describes several approaches and includes many example applications.

In Section 3, preconditioning approaches for symmetric and positive definite matrices are discussed. There follows in Section 4 some brief comments on symmetric semidefinite matrices. This is followed in Section 5 by preconditioning for symmetric and indefinite matrices, and finally for nonsymmetric matrices in Section 6. We start, however, with a brief review of iterative methods in order to orient ourselves.

2. Krylov subspace iterative methods

For any vector $r \in \mathbb{R}^n$, if it is easy to compute the matrix–vector product Ar, then it is easy to generate vectors in the *Krylov subspaces*

$$\mathcal{K}_k(A, r) = \operatorname{span}\{r, Ar, A^2r, \dots, A^{k-1}r\}, \quad k = 1, 2, \dots$$

by forming $A(Ar)$, *etc.* In particular, when A is sparse (*i.e.*, has few non-zero entries in any row), then such matrix–vector products are readily computed. For general dense matrices, a matrix–vector multiplication takes $\mathcal{O}(n^2)$ operations, so there is less possibility for preconditioning to reduce this complexity unless some other structure can be used, such as in the Toeplitz example above.

Krylov subspace iterative methods for finding $x \in \mathbb{R}^n$ satisfying $Ax = b$ compute iterates x_k for which

$$x_k - x_0 \in \mathcal{K}_k(A, r_0), \quad k = 1, 2, \dots \tag{2.1}$$

from some initial guess x_0, and thus require just one matrix–vector product computation at each iteration. The vector $r_0 = b - Ax_0$ is the residual associated with x_0; in general $r_j = b - Ax_j, j = 0, 1, 2, \dots$. If $x_0 = 0$, then

$$x_k \in \mathcal{K}_k(A, b), \quad k = 1, 2, \dots.$$

Thus the iterates and residuals of every Krylov subspace method satisfy

$$x_k - x_0 = \sum_{j=0}^{k-1} \alpha_j A^j r_0$$

for some coefficients $\alpha_j, j = 0, 1, 2, \dots, k-1$. Hence

$$x_k = x_0 + q(A) r_0, \tag{2.2}$$

where q is the polynomial of degree $k-1$ with $q(z) = \sum_{j=0}^{k-1} \alpha_j z^j$. Premultiplying (2.2) by A and subtracting from b, we obtain

$$b - Ax_k = b - Ax_0 - Aq(A) r_0,$$

and thus

$$r_k = r_0 - Aq(A) r_0 = p(A) r_0, \tag{2.3}$$

where

$$p(z) = 1 - z \sum_{j=0}^{k-1} \alpha_j z^j = 1 - \sum_{j=1}^{k} \alpha_{j-1} z^j$$

is a polynomial of degree k which satisfies $p(0) = 1$.

All Krylov subspace methods are thus described by (2.3), and each different Krylov subspace method is characterized by an optimality or (bi-)orthogonality property.

When A is symmetric and positive definite (so that $\|v\|_A = (v^t A v)^{1/2}$ defines a vector norm), the conjugate gradient method (CG) (Hestenes and Stiefel 1952, Algorithm 6.18 in Saad 2003) requires only the one matrix–vector product with A as well as five simple vector operations at each iteration to compute iterates satisfying (2.1), which minimize $\|x - x_k\|_A$ over the Krylov subspace.

When A is symmetric but indefinite, $r^T A r$ takes both positive and negative values, so a norm cannot be defined as for the conjugate gradient method. The Krylov subspace method of choice for symmetric indefinite systems, the minimum residual (MINRES) method (Paige and Saunders 1975, page 187 in Fischer 2011), takes one matrix–vector product with A as well as seven simple vector operations at each iteration to compute iterates satisfying (2.1) which minimize the Euclidean norm of the residual, $\|r_k\|_I = (r_k^T r_k)^{1/2}$.

When A is non-symmetric, there is not such an obvious method of choice, hence several Krylov subspace methods are widely used. The most popular is GMRES (Saad and Schultz 1986, Algorithm 6.9 in Saad 2003) which, similarly to MINRES, computes iterates that minimize the Euclidean norm of the residual, but by contrast to MINRES requires an increasing amount of computation and storage at each successive iteration to achieve this. Thus GMRES can be a good method if only a few iterations are needed to achieve acceptable convergence – this might be the case if one has a good preconditioner – but it is not practical if many iterations are required. Restarting GMRES at (regular) intervals reduces the computational burden, but also in most cases severely slows convergence. Thus use is made of Krylov subspace methods for non-symmetric A which require only a fixed amount of work at each iteration, but which necessarily therefore have more erratic convergence (Faber and Manteuffel 1984). These are generally based on the non-symmetric Lanczos method: we mention BICGSTAB (van der Vorst 1992), BICGSTAB(ℓ) (Sleijpen and Fokkema 1993), QMR (Freund and Nachtigal 1991), IDR(s) (Sonneveld and van Gijzen 2008), though there are many others. There are also several variants; see Simoncini and Szyld (2007).

For any non-singular non-symmetric matrix, it is always possible to attack the symmetric and positive definite system of *normal equations*,

$$A^T A x = A^T b \tag{2.4}$$

(or $AA^T y = b, x = A^T y$), and in some instances this has been proposed; see for example Laird and Giles (2002). A related approach is to solve a symmetric indefinite system in a form such as

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{2.5}$$

However, whichever form is employed, convergence can be rather slow: A^{-T} is not generally such a good preconditioner for A! To give an idea of the convergence that might be expected, if A were already symmetric and positive definite, then generally the number of CG iterations required for (2.4) would be approximately the square of the number of CG iterations for the system $Ax = b$. There are situations in which this may not be such a difficulty, but it is generally harder to find good preconditioners for $A^T A$ (or AA^T) than for A, and this may be the main reason why the non-symmetric Krylov subspace iterative methods above attract more attention and are more widely used. An important and underappreciated general observation in this regard is due to Braess and Peisker (1986).

Remark. P^TP can be an arbitrarily bad preconditioner for A^TA regardless of the quality of P as a preconditioner for A.

The above remark applies even if A, P are symmetric, as is shown by the elementary example in Braess and Peisker (1986). In Section 5 we will see situations in which one can build a particular preconditioner P for A such that P^TP is also a good preconditioner for A^TA. Indeed, the main result in Braess and Peisker (1986) provides sufficient conditions for this to occur.

Now, CG for (2.4) will generate Krylov subspaces of the form $\mathcal{K}(A^TA, A^Tb)$ for example, when $x_0 = 0$. Perhaps the most suitable implementation algebraically equivalent to a symmetric Krylov subspace method for the normal equations is the LSQR algorithm of Paige and Saunders (1982), which also computes least-squares solutions for rectangular A. An advantage of these methods is that the convergence theory of CG and MINRES apply.

The reason for there being definite methods of choice (CG, MINRES) for symmetric matrices, but a whole range of possibilities for non-symmetric matrices, arises because of convergence guarantees. In the case of symmetric matrices there are descriptive convergence bounds which depend only on the eigenvalues of the coefficient matrix, and thus the number of iterations which will be needed for convergence to a given tolerance can be estimated and bounded *a priori*; a good preconditioner will ensure fast convergence. These comments apply also for normal equation approaches to non-symmetric matrices.

By contrast, to date there are no generally applicable *and descriptive* convergence bounds even for GMRES; for any of the non-symmetric methods without a minimization property, convergence theory is extremely limited. The convergence of GMRES has attracted a lot of attention, but the absence of any reliable way to guarantee (or bound) the number of iterations needed to achieve convergence to any given tolerance in most situations means that there is currently no good *a priori* way to identify the desired qualities of a preconditioner. This is a major theoretical difficulty, but heuristic ideas abound.

All Krylov subspace methods are to an extent affected by rounding errors in a computer if run for a sufficiently large number of iterations; see Liesen and Strakoš (2013) or the discussion in Greenbaum (1997, Chapter 4). In this paper we ignore any such effects since an important consequence of good preconditioning is that few iterations should be necessary for acceptable convergence, and thus round-off effects are not usually able to build up and become a significant issue. Similarly, theory exists which describes the eventual (*i.e.*, asymptotic) convergence of Krylov subspace methods such as CG, but this is also not generally relevant to preconditioned iterations: it is rapid error or residual reduction in early iterations that is typically desired.

3. Symmetric and positive definite matrices

The standard CG convergence bound for iterative solution of $Ax = b$, when A is symmetric and positive definite, comes straight from (2.3) and the minimization condition described above, and is given by

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_{\lambda \in \sigma(A)} |p_k(\lambda)|,$$

where $\|y\|_A^2 = y^T A y$, Π_k is the set of real polynomials of degree less than or equal to k, and $\sigma(A)$ is the eigenvalue spectrum of A (Greenbaum 1997, Section 3.1). One immediately deduces that if A has a small number of distinct eigenvalues, say s, then CG will terminate at the sth iteration with the correct solution, since there is a degree s polynomial which has these eigenvalues as its roots. Iteration error can already be acceptably small after only a few iterations when the eigenvalues are clustered. In particular, if one considers all eigenvalues to lie in a single interval $\lambda_{\text{min}}^A \leq \lambda \leq \lambda_{\text{max}}^A$, then use of Chebyshev polynomials yields the convergence bound

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k, \tag{3.1}$$

where $\kappa = \kappa^A = \lambda_{\text{max}}^A / \lambda_{\text{min}}^A$ is the Euclidean norm (or 2-norm) condition number for such a symmetric positive definite matrix; see again Greenbaum (1997, Section 3.1). An equivalent statement is that it takes approximately $\frac{1}{2}\sqrt{\kappa}|\log \epsilon/2|$ iterations to ensure that $\|x - x_k\|_A / \|x - x_0\|_A \leq \epsilon$.

The origin of the name *preconditioning* is generally now associated with this convergence bound in the sense that reduction of the condition number, κ, guarantees faster convergence (though for historical accuracy, see Benzi 2002); what we now understand as preconditioning is, however, a much broader concept. If P is also symmetric and positive definite and the extreme generalized eigenvalues of the 'pencil' $Av = \lambda P v$ (equivalently the extreme eigenvalues of $P^{-1}A$) satisfy

$$\kappa^A = \frac{\lambda_{\text{max}}^A}{\lambda_{\text{min}}^A} \gg \frac{\lambda_{\text{max}}^{P^{-1}A}}{\lambda_{\text{min}}^{P^{-1}A}} = \kappa^{P^{-1}A},$$

then CG convergence for the preconditioned system is guaranteed to occur in far fewer iterations than for the original unpreconditioned system, since (3.1) now holds with $\kappa = \kappa^{P^{-1}A}$. Note that for CG even with preconditioning, it remains the relative magnitude of the error in $\|\cdot\|_A$ which arises in the bound (3.1); a symmetric positive definite preconditioner affects the eigenvalues and thus the speed of convergence, but not the norm in which convergence occurs. Use of a non-symmetric preconditioner, P, would prevent use of CG; in that case a non-symmetric Krylov subspace method would have to be used in general (but see Section 7).

To establish the quality of P as a preconditioner for A, it is sufficient to find (and *a priori* prove if possible!) upper and lower bounds on the eigenvalues of the form

$$\ell \leq \lambda_{\min}^{P^{-1}A}, \quad \lambda_{\max}^{P^{-1}A} \leq \Upsilon \tag{3.2}$$

or equivalently – and usually more tractably – upper and lower bounds on the *generalized Rayleigh quotient*

$$\ell \leq \frac{u^T A u}{u^T P u} \leq \Upsilon, \quad \text{for all } u \in \mathbb{R}^n - \{0\}. \tag{3.3}$$

In either case $\kappa^{P^{-1}A} \leq \Upsilon/\ell$ leads directly to a convergence bound through (3.1) since the right-hand side varies monotonically with κ.

One does not actually replace A with $P^{-1}A$ in the CG algorithm to implement preconditioning in the case of symmetric A, since this would generally destroy symmetry, but even when symmetry is preserved (as in Saad 2003, Algorithm 9.1), the action of P^{-1} is required on a different vector at each iteration.

One thus has clear sufficient criteria for identifying a good preconditioner, P, in the symmetric positive definite case:

- it should be easy to solve systems of equations with the preconditioner P as coefficient matrix, that is, the action of P^{-1} on a vector should be readily computed,
- $\kappa^{P^{-1}A}$ should be small.

These are competing aims: $P = I$ satisfies the first but not the second unless A is already well-conditioned, and $P = A$ satisfies the second but not the first unless solution of systems with A is already easy!

3.1. Diagonal scaling

The earliest preconditioning ideas came from consideration of the first of these criteria, and are largely in the 'given a matrix' category. Taking P to be a diagonal matrix is very simple; if in fact $P = \mathrm{diag}(A)$, then this is widely called Jacobi preconditioning. For a well-conditioned matrix that is just badly scaled, this is generally all that is needed, but in general diagonal preconditioning is likely to achieve very little in terms of reducing iterations or computation time except in specific situations as below. There are two related types of problem where Jacobi preconditioning (or just diagonal scaling) is important.

Firstly, the 'identity' matrix for finite element computation is the *mass matrix*, Q, which is the Gram matrix of the finite element basis functions

$\{\phi_i, i = 1, \ldots, n\}$, that is, it has entries

$$Q = \{q_{i,j}, i, j = 1, \ldots, n\}, \quad q_{i,j} = \int_\Omega \rho \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) \tag{3.4}$$

for a domain Ω. Here the potentially variable coefficient ρ might, for example, represent material density in a structure, and if this is highly variable or if the discretization has a large variation in the size (length, area, volume) of the elements – as is inevitable on an irregular or adapted mesh – then this variability is also represented in the diagonal entries of Q. Thus $P = \operatorname{diag}(Q)$ is provably a good preconditioner in this case (Wathen 1986). For example the `gallery('wathen',n,n)` matrix in MATLAB is a finite element mass matrix with randomly variable density for a so-called serendipity finite element. The precise eigenvalue bounds of the form (3.2) or (3.3), together with tabulated values for ℓ and Υ for many types of finite elements, are derived in Wathen (1986).

We briefly comment that in problems as above with analytic *a priori* eigenvalue bounds ℓ and Υ which are (reasonably) tight, the possibility also arises to use a fixed but small number of Chebyshev (semi-)iterations as a (linear) preconditioner; see Wathen and Rees (2009).

The second important situation arises also through highly variable coefficients; the classic example arises from the operator

$$\nabla \cdot (K \nabla u),$$

where the permeability K has large jumps. For continuous finite element approximation of such terms, Graham and Hagger (1999) were the first to identify the utility of diagonal scaling to render the preconditioned problem more homogeneous. For such an elliptic PDE, Jacobi preconditioning is highly unlikely to be an adequate preconditioner on its own, but when used together with an appropriate preconditioner for a homogeneous elliptic PDE operator (see below), the combination can be much better than just the elliptic preconditioner by itself. In the literature there are several developments of the idea of Graham and Hagger, including for non-isotropic permeabilities.

3.2. Incomplete Cholesky factorization

It is well known that Cholesky factorization (LL^T or LDL^T factorization where L is lower triangular and D is diagonal) of a symmetric and positive definite matrix leads immediately via forward and backward substitution to direct solution of a matrix system. For sparse matrices, unfortunately, the factor L is usually much less sparse (*i.e.*, it has many more non-zeros) than A. Such fill-in is usually unavoidable, though widely used sparse direct methods use sophisticated algorithms in attempting to minimize it (HSL 2013, Davis 2004, Amestoy, Duff, L'Excellent and Koster 2000, Li 2005).

In the context of PDE problems, the essential rule is that a sparse direct method can be a good solver for a problem on a two-dimensional spatial domain, but is generally computationally infeasible for a three-dimensional problem. If one preselects (or dynamically selects) the sparsity pattern of L and drops terms which would fill outside this pattern, then one can form an *incomplete* Cholesky factorization

$$A = LL^T + E,$$

where E represents the error due to the dropped terms. One can additionally *threshold* so that small entries are not kept in the incomplete factors. With $P = LL^T$, it is clear that forward and backward substitution still enable easy solution of systems with the preconditioner, and the main question remaining is whether the preconditioner leads to fast convergence of CG.

In general, many different variants of this basic idea exist, and software for various incomplete factorization-based algorithms is widely available. This is certainly a plus-point, but unless A is diagonally dominant, incomplete Cholesky algorithms do not always work well, indeed they do not always exist without modification; see Meijerink and van der Vorst (1977), and see Benzi (2002) for discussion of this as well as many other aspects of incomplete factorization preconditioning. With regard to robustness, there are certain advantages to computing an incomplete triangular factorization not using Cholesky but via orthogonalization in the inner product generated by A; see Benzi and Tuma (2003). This approach is called RIF (robust incomplete factorization) and shares some similarity with the AINV sparse approximate inverse discussed below.

Let us make some of these statements precise in the context of solving a simple discretized elliptic boundary value problem, namely the common 5-point finite difference approximation of the Dirichlet problem for the (negative) Laplacian

$$-\nabla^2 u = f \quad \text{in } \Omega, \quad u \text{ given on } \partial\Omega,$$

which arises from second-order central differencing on a regular grid with spacing h in each of the two spatial dimensions. The choice of sign ensures that the operator and thus the arising matrix are positive rather than negative definite. If the domain, Ω, is the unit square (so the boundary $\partial\Omega$ is four coordinate-aligned line segments), then the arising matrix A is of dimension $N \approx h^{-2}$ for small h and also $\kappa^A = \mathcal{O}(h^{-2})$; see Elman *et al.* (2014*b*, Section 1.6). Thus (3.1) predicts that $\mathcal{O}(h^{-1})$ CG iterations would be required for convergence of unpreconditioned CG. The simplest incomplete Cholesky preconditioner for which L has the same sparsity pattern as the lower triangular part of A also leads to $\kappa^{P^{-1}A} = \mathcal{O}(h^{-2})$, though there is significant clustering of a large number of the eigenvalues around 1; the bound (3.1) based only on the extreme eigenvalues is pessimistic in this situation. A variant, modified incomplete factorization (due to Gustafsson 1978 but related to rather earlier work by Dupont, Kendall and Rachford 1968) reduces this to $\kappa^{P^{-1}A} = \mathcal{O}(h^{-1})$, and generally leads to slightly faster CG convergence. Applying (3.1) in this case predicts that $\mathcal{O}(h^{-1/2})$ preconditioned CG iterations would be required for convergence, and this is close to what is seen in practical computation. This implies that for a sequence of such problems with smaller and smaller h, it will take an increasing number of iterations to achieve acceptable convergence as h gets smaller (and so the matrix dimension gets larger) – a rather undesirable feature.

Fortunately, there are much better preconditioners for this and related elliptic PDE problems which guarantee $\kappa^{P^{-1}A} = \mathcal{O}(1)$, so that the number of required CG iterations is essentially constant for larger-dimensional discrete problems coming from approximation on finer grids. Such preconditioning methods are said to have P *spectrally equivalent* to A. The most important class of such methods arises from multigrid ideas, which we consider next.

Of course, even in the more general context of problems not coming from PDEs, incomplete Cholesky and its variants can be of considerable utility; in particular, algebraic preconditioners of this form can very often be applied.

Before moving on, we comment that preconditioning techniques have been introduced that turn out to be related to incomplete factorization. A particular example is the 'support graph' preconditioner introduced originally in a conference talk by Vaidya, who wrote commercial software but did not publish on the subject. The support graph idea comes out of graph-theoretic notions and has a somewhat discrete mathematical flavour. Later analysis both clarified the scope of the ideas – they are now essentially extended to cover most symmetric and diagonally dominant matrices – and made connections with incomplete factorization; see Bern *et al.* (2006), Spielman and Teng (2014) and references therein.

3.3. Multigrid

Since the groundbreaking paper by Brandt (1977), multigrid methods have been at the core of scientific computing. Earlier contributions, notably by Fedorenko (1964) and Bakhvalov (1966), are now considered to have introduced the key ideas of combining a simple smoothing iteration (such as Jacobi or Gauss–Seidel iteration) with coarse grid correction on successively coarser grids for the finite difference Laplacian, but Brandt's vision was much broader: he showed just how fast multigrid solvers could be and introduced many ideas to keep researchers busy for decades! For a positive definite elliptic PDE operator, it has been clear for a long time that an appropriate multigrid method will provide an optimal solver, that is, a solver such that the work to compute the solution scales linearly with the dimension of the discretized problem.

Classically, multigrid is a stationary iteration (albeit a non-trivial one) for solving $Ax = b$ when A is a discrete Laplacian matrix (arising from finite difference or finite elements). For V- or W-cycles, iterates satisfy single-cycle contraction of the form

$$\|x - x_k\|_A \leq \eta \, \|x - x_{k-1}\|_A, \tag{3.5}$$

where the contraction factor, η, is typically around 0.1 (see, *e.g.*, Elman *et al.* (2014*b*, Section 2.5)). It follows that the linear multigrid operator (represented by a complicated and fairly dense matrix which one would never want to use in computations) is an excellent candidate for a preconditioner, since from (3.5) it readily follows that

$$1 - \eta \leq \frac{x^T A x}{x^T P x} \leq 1 + \eta$$

(Elman *et al.* 2014*b*, Lemma 4.2). That is, applying one multigrid V-cycle, say, is equivalent to solution of a preconditioner system with preconditioner P which guarantees rapid CG convergence because $\kappa^{P^{-1}A} \leq (1+\eta)/(1-\eta)$. There is little point in using multigrid cycles as a preconditioner for CG in this way for simple problems: one might as well simply use multigrid iteration alone. For more complicated problems, however, the observation that a single multigrid cycle is an efficient spectrally equivalent approximation of an elliptic PDE operator such as the Laplacian is very useful; see Section 5. Note also the important point that arises here for the first time in this review: a preconditioner can be a linear operation rather than specifically a known matrix. Thus {3 V-cycles}, for example, is a suitable preconditioner. On the other hand, iterating multigrid (or any other iteration) to some residual tolerance will probably ensure that different numbers of V-cycles are used at different Krylov subspace iterations and so the preconditioner is not a fixed linear operator, and the above convergence analysis cannot apply.

Multigrid – and multilevel methodology more generally – has now become a huge subject (see for example Trottenberg, Oosterlee and Schüler 2000), with specialized techniques for the treatment of several specific features which might arise in a PDE (or other) problem. In the past couple of decades, algebraic multigrid methods (AMG), which try to mimic a geometric multigrid approach but use only information from the entries of a matrix rather than geometric information regarding grids, boundary conditions, *etc.*, have attracted much attention, and there is now widely available software, for example HSL_MI20 (Boyle, Mihajlovic and Scott 2010), ML (Gee *et al.* 2006) and BoomerAMG (Henson and Yang 2000). In particular, in situations where a problem is fundamentally of the character of an elliptic differential operator but there are complicating features, an AMG cycle can be an excellent preconditioner. This can be true even if AMG is non-convergent, since it may contract on all but a few modes; the few corresponding outlier eigenvalues are easily dealt with by CG. It can simply be much more convenient to employ an AMG cycle as (part of) a preconditioner for a Krylov subspace method than to have to 'hard-code' bespoke techniques for each different variant problem that comes up. So common is this combined use of AMG and a Krylov subspace method that it is rarely even mentioned that AMG is used as a preconditioner, but this is almost always the case in practice.

We should mention that multigrid ideas have been tried in contexts other than just for PDE problems; see for example Giles (2008) and Livne and Brandt (2012).

It should also be mentioned in this section that a multigrid V- or W-cycle can usually be quite easily made into a symmetric linear operator (see Elman *et al.* 2014*b*, page 98), though the so-called full multigrid cycle, for example, cannot. Positive definiteness is not usually an issue for an underlying positive definite problem.

3.4. Domain decomposition

For a very large-dimensional problem, it might be a fairly natural idea to split into separate subproblems and solve each separately (and usually in parallel) to give an approximation to the solution of the overall problem. In the original context of solving an elliptic boundary value problem on a domain Ω, this is the basic idea of domain decomposition: the subproblems being problems expressed on subdomains $\Omega_k, k = 1, \dots, N$ with $\cup\Omega_k = \Omega$. In its simplest form, it corresponds to a block diagonal preconditioner with each separate subproblem represented by a single diagonal block which may or may not be overlapping, that is, which may or may not share some variables with other subproblems. An important issue is how to treat the variables either in the overlap regions or on the introduced boundaries between subdomains. Domain decomposition was first employed abstractly by Hermann Schwarz in the nineteenth century to prove an existence result for an elliptic PDE, hence methods of this type are often called Schwarz methods.

Like multigrid, domain decomposition is a huge subject which we have no intention of covering with any depth here; see for example the books by Quarteroni and Valli (1999), Smith, Bjørstat and Gropp (1996), Toselli and Widlund (2004), Mathew (2008) or the earlier survey by Chan and Mathew (1994). What is clear, however, even from this brief description is that domain decomposition gives rise to preconditioners that are well suited to parallel computation.

To give a brief idea of what can be achieved on a simple problem, we consider the Poisson equation on a domain $\Omega \subset \mathbb{R}^d$ approximated by standard (conforming) finite elements or finite differences, with a mesh spacing h. If Ω is split into non-overlapping subdomains of width H, then there are basic block diagonal preconditioners without a coarse grid solve for which one can establish that

$$\kappa^{P^{-1}A} \leq CH^{-2}(1 + \log^2(H/h)).$$

The more sophisticated domain decomposition schemes which also include a coarse grid (or 'wire basket' solve) can achieve

$$\kappa^{P^{-1}A} \leq C(1 + \log^2(H/h)).$$

In either case, C is a generic constant not dependent on H, h, and the bounds translate into convergence estimates via (3.1). The size H can be considered to be the coarse grid size in essence; the bounds above assume exact solution of the problems on subdomains.

With the importance of parallel and distributed computing, the domain decomposition concept remains very important and much is known for more advanced methods and for different problems; see the regular *Domain Decomposition* conference proceedings available at www.ddm.org (2014).

3.5. Sparse approximate inverses

For any matrix norm, minimization of

$$\|I - AR\|$$

over all matrices $R \in \mathbb{R}^{n\times n}$ will clearly yield $R = A^{-1}$. Minimization with restrictions on the entries where R may have non-zero values leads to *sparse approximate inverses.* That is,

$$R = \text{argmin}_{R\in S_R} \|I - AR\|,$$

where S_R describes the subset of entries of R which are allowed to be non-zero, is a sparse approximate inverse for A. To be consistent with the nomenclature in this article, $P = R^{-1}$ is the preconditioner here; solution of systems with P is, of course, simply multiplication by R in this situation. S_R is strictly a subspace of matrices with restricted sparsity pattern.

The Frobenius norm

$$\|B\|_F^2 = \sum_{i=1,\ldots,n} \sum_{j=1,\ldots,n} b_{i,j}^2$$

is by far the most widely considered, since the minimization problem then reduces to least-squares problems for the columns of R separately (thus computable in parallel). Further, these least-squares problems are all of small dimension when the specification of S_R ensures that R is a sparse matrix. We again refer to Benzi (2002) for a thorough description.

As posed in this simple form, the sparse approximate inverse, R, is not generally symmetric when A is symmetric. However, we introduce this type of preconditioner in this section because symmetry can be enforced by computing R in factored form $R = LL^T$, where L is sparse and lower triangular; see Kolotilina and Yeremin (1993). Symmetry of a sparse approximate inverse can also be achieved by the AINV algorithm of Benzi, Meyer and Tuma (1996) and the more robust (stabilized) SAINV variant (Benzi, Cullum and Tuma 2000), which is guaranteed not to break down if A is positive definite. See again Benzi (2002).

The main issue in the construction of any sparse approximate inverse is the choice of the sparsity pattern S_R. The simplest idea is to use the sparsity of A, whilst there have been several attempts to dynamically define S_R, most notably in the SPAI algorithm of Grote and Huckle (1997). An extension which can be used in a number of different scenarios is to use a *target matrix*, T, and hence minimize $\|T - AR\|_F$; see Holland, Wathen and Shaw (2005). A further comment is that sparse approximate inverse algorithms are clearly quite widely applicable, but generally do not give spectrally equivalent preconditioners for elliptic PDEs. They do, however, make excellent smoothers for use with multigrid (Tang and Wan 2000, Bröker, Grote, Mayer and Reusken 2001).

3.6. Hierarchical representations

For symmetric and positive definite matrices arising via low-order finite element approximation from elliptic PDEs, Yserentant (1986) introduced the hierarchical basis preconditioner nearly three decades ago. This uses the idea that the coefficient matrix would have a much smaller condition number if a different (but less computationally convenient) basis for the approximation space were employed. By using a hierarchical basis – which would give rise to a fairly dense matrix – both this good conditioning *and* a rapid change of basis matrix based on a simple tree structure allow the sparse matrix arising from the usual locally defined finite element basis functions to be effectively preconditioned.

For the discrete Laplacian on a two-dimensional domain, Yserentant's hierarchical preconditioner reduces $\kappa = \mathcal{O}(h^{-2})$ to $\kappa = \mathcal{O}(\log h^{-1})$ for the discrete Laplacian (see Yserentant 1986 or Elman and Zhang 1995 for a more linear algebraic proof). Unfortunately, for three-dimensional domains only $\kappa = \mathcal{O}(h^{-1})$ is possible with the straightforward use of the hierarchical basis (Ong 1997), though an improved variant of this idea due to Bramble, Pasciak and Xu (1990) can yield $\kappa = \mathcal{O}(1)$ in three dimensions as well. These ideas, originally introduced as based on the multilevel splitting of finite element spaces, were the precursor to a whole abstract theory based on such splittings; the review article by Xu (1992) remains a largely comprehensive description of this abstract theory, which covers many approaches including multigrid and domain decomposition discussed above for symmetric and positive definite matrices arising from elliptic PDEs.

A different hierarchical structure can also be of utility. We wish to mention some ideas and the associated computational technology which enables certain dense matrices – such as the inverse matrices of discrete positive definite elliptic PDE operators – to be represented to machine accuracy in a highly efficient format. There are a number of related but differing approaches (Chandrasekaran *et al.* 2005, Greengard and Rokhlin 1987, Hackbusch 1999, Grasedyck and Hackbusch 2003, Hackbusch, Khoromskij and Sauter 2000, Bebendorf 2008), but the essential observation can be thought of in terms of the Green's function for an elliptic operator: nearby entries need accurate representation, but variables distant from a given entry can be treated together. This notion is most directly represented in the $\mathcal{H}$-matrix representation of Hackbusch: off-diagonal blocks are represented as low-rank blocks using outer products of vectors or matrices with very few columns. By employing a blocking hierarchy, the computational complexity of storing (and also multiplying with) such matrix approximations can be as low as $\mathcal{O}(n \log n)$.

We mention these ideas here since such approximate representations of dense matrices have been suggested as preconditioners rather in the same vein as sparse approximate inverses. They can also be utilized in other more demanding situations, as we describe later.

4. Semidefinite matrices

It might seem rather odd to be interested in preconditioners for singular matrices; one might believe that only well-posed problems were worthy of attention. However, there are different situations in which approximations to certain singular operators/matrices are valuable and we mention some ideas in this brief section.

The simplest PDE problem which gives rise to a singular matrix is probably the Neumann problem for a simple operator such as the Laplacian; specifying only derivative boundary conditions for a potential-like variable leaves a one-dimensional kernel comprising the constants, as is well known. Such a small-dimensional null space generally causes little significant difficulty in terms of iterative methods or preconditioning, and methods for definite matrices generally apply; see Elman *et al.* (2014*b*, Section 2.3). Similarly, graph analysis and Markov chain problems, respectively, lead to symmetric and non-symmetric singular systems with simple, low-dimensional kernels; again, methods for definite matrices should generally be applied without explicit 'fixes' to ensure non-singularity for such problems. For problems arising in contexts other than PDEs, singularity tends to imply that a description has not been completely specified.

The issue of semidefiniteness is certainly serious, however, for equations involving the curl or div operators applied to vector fields, as arise, for example, in the Maxwell equations describing electromagnetics. In this context, the operator curl curl has all gradients of scalar fields in its kernel, and thus under most discretizations one would expect that the matrices arising will be semidefinite with a kernel of some significant dimension. Similarly, the curl of any smooth enough vector field is in the kernel of the div operator. On the relevant spaces of functions, the curl curl operator is, however, positive definite and self-adjoint (symmetric), and it seems that it may be possible to use convenient preconditioners such as standard AMG implementations if appropriate account is taken of the kernel. This is the purpose of 'auxiliary space' preconditioning, which aims to take account of the natural decompositions of function spaces that arise. The ideas have evolved over some years; see Hiptmair and Xu (2007) and references therein. The main practical issue is representing mapping operators between the various spaces in the discrete setting.

5. Symmetric and indefinite matrices

For real symmetric matrices which do not have to be definite, the Krylov subspace method of choice is the MINRES algorithm due to Paige and Saunders (1975). In the same paper they introduced the alternative but less used SYMMLQ method. As its name implies, MINRES minimizes the norm of the residual vector for x_k in the k-dimensional Krylov subspace at each iteration. Thus, since the symmetry of A ensures that the eigenvalues are real and the eigenvectors are orthogonal, the convergence of MINRES is described by

$$\frac{\|r_k\|_I}{\|r_0\|_I} \leq \min_{p_k\in\Pi_k, p_k(0)=1} \max_{\lambda\in\sigma(A)} |p_k(\lambda)|, \tag{5.1}$$

where the notation is as used before in the positive definite case. The difference from that case is not only that the residual rather than the error is minimized (it is really a difference of norm since $r_k = A(x-x_k)$ implies that $\|r_k\|_I = \|x - x_k\|_{A^2}$), but the spectrum, $\sigma(A)$, now contains both positive and negative eigenvalues.

As in the symmetric and positive definite case, (5.1) immediately indicates that termination of the iteration will occur with the solution at the sth iteration if A has just s distinct eigenvalues. Correspondingly, the eigenvalues lying in a small number of clusters should also lead to rapid convergence if none of the clusters is too far from or too close to the origin. If the eigenvalues of A are all contained in two intervals

$$[-a, -b] \cup [c, d],$$

where a, b, c, d are all positive, then it is less straightforward in general than in the positive definite case to derive a simple convergence bound (see Wathen, Fischer and Silvester 1995). One situation where a simple expression is possible, however, is when $d - c = a - b$; the MINRES residuals then satisfy the convergence bound

$$\frac{\|r_{2k}\|_I}{\|r_0\|_I} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \tag{5.2}$$

where $\kappa = ad/(bc)$ (see Elman *et al.* 2014*b*, Section 4.2.4). Note that κ here plays a similar role to $\lambda_{\max}/\lambda_{\min}$ in the positive definite case, though it is not the matrix condition number here; since the Euclidean norm condition number, $\|A\|\|A^{-1}\|$, is $\max\{a, d\}/\min\{b, c\}$ in this situation, κ can be as large as the square of this condition number when $a = d, b = c$, but rather less when the two intervals are not symmetrically placed about the origin. When $a = c$, for example, $\kappa = d/b$ is again exactly the Euclidean norm condition number.

When A is symmetric and indefinite, any preconditioner for MINRES must be symmetric and positive definite. This is necessary since otherwise there is no equivalent symmetric system for the preconditioned matrix. Thus a nonsymmetric iterative method must be used even if a symmetric and indefinite preconditioner is employed for a symmetric and indefinite matrix in general. An indefinite preconditioner is therefore not usually desirable, though there are exceptions: see below. A preconditioner for a symmetric indefinite matrix A for use with MINRES therefore cannot be an approximation of the inverse of A, since this is also indefinite. This simple observation rules out several approaches described above in the positive definite case.

With a symmetric and positive definite preconditioner, the preconditioned MINRES convergence bounds above become

$$\begin{aligned}\frac{\|r_k\|_{P^{-1}}}{\|r_0\|_{P^{-1}}} &\leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_{\lambda \in \sigma(P^{-1}A)} |p_k(\lambda)| \\ &\leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_{\lambda \in [-a,-b]\cup[c,d]} |p_k(\lambda)|,\end{aligned} \tag{5.3}$$

where $[-a, -b] \cup [c, d]$ is now an inclusion set for all of the eigenvalues of $P^{-1}A$ rather than A. A further aspect introduced by preconditioning for MINRES, which does not arise for CG in the positive definite case, is now apparent: the norm in which convergence naturally occurs is affected by the preconditioner, so it is not just the eigenvalue distribution and hence the speed of convergence that is affected.

The only reason we bring up this issue is because one now has the undesirable possibility that a preconditioner which apparently gives rapid MINRES convergence might give inaccurate solution iterates simply because

$\|\cdot\|_{P^{-1}}$ is a highly distorted norm. The following trivial example will make the point.

Let A be any indefinite diagonal matrix and suppose we precondition with

$$P = \operatorname{diag}(\epsilon, 1/\epsilon, \dots, 1/\epsilon)$$

for some small positive ϵ. Convergence to some small tolerance will appear to occur after only one iteration, since there is only one large entry of P^{-1} and the remaining entries are very small. Because of this distorted weighting, the first component of the first solution iterate will be accurate, but the rest will generally not be. For further consideration of this issue see Wathen (2007).

These different issues, which do not arise when A is symmetric and positive definite, lead some authors to the conclusion that preconditioning for symmetric indefinite matrices is much more difficult than the definite case. However, it is still much more tractable than the non-symmetric case! For a PDE problem, for example, the convergence bound (5.3) guarantees that preconditioned MINRES convergence will be in a number of iterations independent of any mesh size parameter, h, given a preconditioner which ensures that a, b, c, d are bounded, bounded away from zero and independent of h. This is like the situation for spectrally equivalent preconditioners in the symmetric positive definite case. Such guarantees rarely exist for non-symmetric iterative methods; see Section 6 below.

We mention that the SQMR Krylov subspace iterative method (Freund and Nachtigal 1994) is designed for indefinite matrices with indefinite preconditioning. This method, however, has no convergence guarantees, being a variant of the non-symmetric QMR iterative method, for which symmetry allows a more streamlined implementation.

General preconditioning strategies for symmetric indefinite matrices are not so readily available as in the positive definite case. Amongst the ‘given a matrix’ approaches, simple application of scaling and incomplete factorization are not contenders and there is little significant work on sparse approximate inverses since these appear unlikely to provide much benefit, except possibly when A has only a few negative eigenvalues. ‘Given a problem’ is more promising, and there are two general classes of problem which give rise to symmetric indefinite matrices for which helpful – sometimes excellent – preconditioning approaches are known.

5.1. Shifted elliptic operators: Helmholtz problems

It might seem that taking a symmetric and positive definite matrix, K, and simply shifting it by subtracting some multiple of the identity should lead to matrices for which easy solution algorithms exist. In general, however, this is not the case. Certainly the matrix $K - \sigma I$ can have resonances (and thus be singular) when σ is an eigenvalue of K, but the difficulty lies deeper than just this observation.

The classic (and important) PDE problem here is the Helmholtz equation $-\nabla^2 u - \sigma u = f$, which arises when looking for periodic (wave-like) solutions of the linear wave equation; the parameter σ is then related to the frequency. Supposing that K is the discretized form of the positive definite operator $-\nabla^2$ (*i.e.*, the negative Laplacian), then for $\sigma > \lambda_{\min}(K)$ it is clear that

$$A = K - \sigma I \tag{5.4}$$

is indefinite and symmetric. Many of the important physical applications involve high frequencies and thus large values of σ. Having a good preconditioner for K – we have identified many in the sections above – is of little help in constructing a preconditioner for A. For finite element discretization, the identity operator becomes the mass matrix Q (3.4), but via diagonal approximation or since the grids employed are generally fairly regular for such wave problems, there is little lost in description by considering (5.4).

In fact, for Helmholtz problems there is usually a radiation boundary condition for outgoing waves or an absorbing boundary condition (or 'perfectly matched layer') is applied so that there is little reflection of outgoing waves back into the computational domain. These add some complex entries in the matrix which generally render it complex and symmetric (not Hermitian). Nevertheless, it is the indefiniteness of Helmholtz problems which remains the most important feature, hence why we consider them in this section. This aspect can, however, mean that non-symmetric Krylov subspace iterative methods are required; SQMR is quite popular in this context.

Magolu Monga Made, Beauwens and Warzée (2000) and Laird and Giles (2002) were perhaps the first to recognize the potential of preconditioning (5.4) with a matrix of the form $K + \mu I$ where μ is positive. In the case of real μ, such a positive definite matrix is readily approximated by standard multigrid cycles which can be used as a positive definite preconditioner for (5.4). Complex values for μ have also been explored (see also van Gijzen, Erlangga and Vuik 2007, for example), and good choices of the parameter can be found from Fourier analysis of the spectrum of the preconditioned operator/matrix. As well as indicating the speed of Krylov subspace iteration convergence, such analysis can also assist in the choice of multigrid components (Erlangga, Oosterlee and Vuik 2006). Elman, Ernst and O'Leary (2001) also propose a multigrid approach for Helmholtz problems employing GMRES as a smoother. Domain decomposition approaches have also been suggested (Gander, Magoulès and Nataf 2002).

A more recent proposal by Engquist and Ying (2011*a*) for variable and high-frequency regimes involves 'sweeping' from the absorbing boundary, eliminating the variables locally in layers. The resulting preconditioner is thus a block incomplete LDL^T factorization. A key aspect of the implementation is that hierarchical ($\mathcal{H}$-)matrix compression techniques as described in Section 3 above can be employed for the intermediate layers; this leads to significant reductions in storage and computational work and leads to excellent overall complexities such as $\mathcal{O}(n\log^2 n)$ for computation of $P = LDL^T$ and application of P^{-1} to a vector for a two-dimensional domain problem. Krylov subspace iterative convergence is observed to occur in few iterations, making this a very competitive technique. Unfortunately, although the idea – and a closely related one employing frontal sparse factorization (Engquist and Ying 2011*b*) – is observed to work well for three-dimensional domains as well, the $\mathcal{H}$-matrix compression does not appear to yield the same small computational complexity: algebraic factors in n replace logarithmic factors in estimates of the complexity for setting up the preconditioner (Poulson, Engquist, Li and Ying 2013). That is, for a three-dimensional domain problem, the computation of $P = LDL^T$ takes time which grows like n^γ for some constant $\gamma > 1$ whereas, once P has been computed, the action of P^{-1} can be computed in $\mathcal{O}(n\log n)$ work at each iteration (similarly to the FFT). For a problem with many different right-hand side vectors there could thus be some amelioration in the work required to set up the preconditioner.

We should alert the reader to the article by Ernst and Gander (2012), which explains the general difficulty of iterative solution of Helmholtz problems.

The Maxwell equations describing electromagnetics have wave-like solutions and hence some of the character of Helmholtz problems. However, they also (in various formulations) have additional semidefinite features, as discussed in the section above, and additional indefinite saddle point structure, and are therefore considered below.

The idea of sweeping through the discrete variables for a problem in a particular order is also important for preconditioner construction for nonsymmetric matrices which derive from PDEs involving advection/convection (or transport), as described below.

5.2. Saddle point systems

The second main class of symmetric and indefinite problems come from minimization of quadratic functionals subject to linear constraints. In fact, linearizations of much more general constrained optimization problems quite generally result in linear systems of the same form,

$$\mathcal{A}x = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{5.5}$$

which are called *saddle point systems*; see Benzi, Golub and Liesen (2005) for a comprehensive treatment of such problems. The block $A \in \mathbb{R}^{n\times n}$ is usually positive definite on the kernel of $B \in \mathbb{R}^{m\times n}$ with $m < n$. This is a sufficient condition for invertibility of $\mathcal{A}$ provided B is of full rank. Often A is positive definite, though in the context of sequential quadratic approximations in optimization (and possibly elsewhere), indefiniteness can arise; see Gould and Simoncini (2009) for spectral bounds in this less usual case. The variables u are the primary variables and p are the Lagrange multipliers. Sometimes the zero block is replaced with a regularization matrix $-C \in \mathbb{R}^{m\times m}$, where C is symmetric and positive semidefinite; if $C \neq 0$, then some authors refer to a regularized or generalized saddle point problem.

Many problems give rise to matrices of this type: as well as the ubiquitous appearance of such problems in optimization, several important PDE problems, including the Stokes problem describing slow viscous incompressible flow, are of this form. The incompressible Navier–Stokes equations give matrices of a similar form but in which A is non-symmetric. For these applications, the primary variables are the fluid velocity and the Lagrange multiplier variables are the fluid pressure. For other applications and references, see again Benzi *et al.* (2005) or Pestana and Wathen (2015*b*).

The appearance of the zero diagonal block in (5.5) causes problems for most of the preconditioning approaches identified in Section 3 above. This is also generally true with regularization, $C \neq 0$. Multigrid approaches, for example, require more complex smoothers; see for example John and Tobiska (2000) and Manservisi (2006). Domain decomposition can be applied, but then the subdomain problems remain indefinite. Incomplete factorization ideas can sometimes be employed if, for example, A and C are diagonal or nearly so (Bergamaschi, Gondzio and Zilli 2004).

A simple observation in the case that A is non-singular is that if

$$P = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \qquad S = BA^{-1}B^T, \tag{5.6}$$

then the preconditioned matrix $P^{-1}\mathcal{A}$ has only three distinct eigenvalues independently of m, n; thus MINRES (or GMRES in the non-symmetric case) will converge in three iterations to the solution (Murphy, Golub and Wathen 2000, Korzak 1999). P in (5.6) is rarely a practical preconditioner for $\mathcal{A}$ in its exact form, but this observation does give a paradigm for the construction of preconditioners and, in particular, highlight the utility of approximations to the *Schur complement*, S. An important point to note is that P is nothing like an inverse of $\mathcal{A}$ (nor an approximation thereof). Nevertheless, good approximations (preconditioners) $\widehat{A}$ for the block A and $\widehat{S}$ for the Schur complement, S, lead to block diagonal preconditioners

$$P = \begin{bmatrix} \widehat{A} & 0 \\ 0 & \widehat{S} \end{bmatrix} \tag{5.7}$$

for $\mathcal{A}$, for which MINRES converges in few iterations. Precisely, if

$$\ell \leq \frac{u^T A u}{u^T \widehat{A} u} \leq \Upsilon \quad \text{and} \quad \gamma \leq \frac{p^T S p}{p^T \widehat{S} p} \leq \Gamma \quad \text{for all } u \in \mathbb{R}^n - \{0\} \text{ and } p \in \mathbb{R}^m - \{0\}$$

for positive constants ℓ, Υ, γ and Γ, then there are positive constants a, b, c and d satisfying the preconditioned MINRES convergence bound (5.3). In the context of PDE problems, for example, this means that preconditioned MINRES is a solver of optimal computational complexity provided that $\widehat{A}$ is spectrally equivalent to A and $\widehat{S}$ is spectrally equivalent to S. A multigrid cycle can be an excellent choice for $\widehat{A}$. See Pestana and Wathen (2015*b*) for a general proof which highlights the role of *inf-sup stability* associated with the minimal singular value of B (in appropriate norms).

For many problems there are *natural* Schur complement approximations which arise from problem formulation and do not require separate approximation of the matrices B and A; see Pestana and Wathen (2015*b*). In the case of the Stokes equations, the natural Schur complement approximation is particularly simple: it is the mass matrix (3.4) (Verfürth 1984, Silvester and Wathen 1994). For any such problem with a natural Schur complement approximation, (5.7) is an excellent preconditioning approach since $\widehat{A}$ can usually be any of the approximations in Section 3. It should be noted that once such approximations $\widehat{A}$ and $\widehat{S}$ are identified, then there are more involved ways to make use of them than the simple block diagonal preconditioner (5.7). When matrix symmetry is not an issue, a block triangular preconditioner can be employed; see Section 6.

Sometimes it is actually useful to directly approximate the matrices B and A in constructing $\widehat{S}$, for example in the context of optimization with constraints which are PDEs (Rees, Dollar and Wathen 2010*a*), though care must be taken to satisfy the sufficient conditions of Braess and Peisker (1986).

For saddle point problems arising particularly in the more general context of optimization, *constraint preconditioning* has attracted significant attention (Keller, Gould and Wathen 2000). Here

$$P = \begin{bmatrix} H & B^T \\ B & 0 \end{bmatrix}$$

is used as a preconditioner for $\mathcal{A}$. Note that the blocks B and B^T which come from the constraints are kept, whilst A is replaced by another symmetric and (usually) positive definite matrix H. It might appear that solution of systems with P may not be significantly easier than with $\mathcal{A}$, but by leaving H implicitly defined, this can be achieved (Dollar and Wathen 2006, Dollar, Gould, Schilders and Wathen 2006). One significant advantage of keeping the constraint blocks in the preconditioner is that the preconditioned problem is now equivalent to a positive definite problem on the constraint manifold, hence a projected CG algorithm can be used (Gould, Hribar and Nocedal 2001); this is rather useful since a constraint preconditioner is symmetric and indefinite, precluding simple use of MINRES. Keller *et al.* (2000) prove that when H is positive definite then $P^{-1}\mathcal{A}$ has $2m$ eigenvalues of 1, each of geometric multiplicity 2 (thus the minimum polynomial generically has a factor $(\lambda - 1)^2$), and that the remaining $n - m$ eigenvalues are real and interlace those of $H^{-1}A$. Thus if H is a good preconditioner for A and systems with P can be readily solved, then this can be an attractive strategy.

Different challenges arise and different strategies are needed for saddle point problems (5.5) with semidefinite block A. The constraint preconditioning paradigm can be applied, in which case finding a matrix H for the semidefinite matrix A may require ideas as in Section 4 above. The matrix $\mathcal{A}$ is still invertible provided A is positive definite on the kernel of B and B is of full rank, as has been mentioned. One possible approach is therefore via an *augmentation* or *augmented system* preconditioner

$$P = \begin{bmatrix} A + B^T W^{-1} B & 0 \\ 0 & W \end{bmatrix}$$

for some appropriate symmetric and positive definite matrix W. This approach is described and analysed by Greif and Schötzau (2006), who were motivated by the solution of Maxwell equations describing electromagnetic phenomena – a key PDE problem with this structure. For related but earlier ideas associated with the semidefiniteness of the div operator, see Arnold, Falk and Winther (1997).

For the various Maxwell equation formulations, semidefiniteness arises through curl curl terms, indefiniteness because of a div constraint, and wave-like solutions are always expected, which means that fine computational grids must be used, especially for high-frequency problems. Fortunately, Maxwell's equations are linear and self-adjoint in most formulations, which is why we bring them up in this section. Mixed finite element approximation is widely used for these PDEs. Representative of the large and still growing literature on preconditioning for such problems are Greif and Schötzau (2007), who develop block preconditioners, and Tsuji, Engquist and Ying (2012), who extend sweeping preconditioner ideas to these problems.

A slightly more detailed review of preconditioning for saddle point problems can be found in Benzi and Wathen (2009).

5.3. A word of caution

As has already been mentioned in Section 2, for any non-singular square matrix A, the solution of a linear system can be achieved by using an indefinite symmetric system of twice the dimension of the form (2.5) or even

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Since the eigenvalues of the coefficient matrix here are exactly symmetric about the origin (they are $\pm\sigma_i(A)$, where $\sigma_i(A)$ are the singular values of A), generically, the convergence of MINRES would stagnate at every other iteration, so the computational effort in solution is exactly the same as for the normal equations (2.4) (Fischer, Ramage, Silvester and Wathen 1998). For (2.5) a similar issue arises.

The artificial construction of indefinite symmetric systems is therefore unlikely to achieve other than equivalent problem statements, except possibly in the solution of least-squares problems by augmented matrix methods (Björck 1996, Chapter 7). To go further, a symmetric indefinite matrix of any form which has eigenvalues symmetrically arranged about the origin will generically lead to *staircasing* behaviour, where progress is made only on alternate MINRES iterations; CG for (2.4) generally requires the same computational effort for equation solution in such circumstances. In this sense, it is highly unfortunate that the tractable MINRES convergence bound (5.2) applies only when the positive and negative eigenvalues extend over intervals of equal length. If in fact $a = d$ and $b = c$, so that these intervals are centred on the origin, then the undesirable staircasing effect is always implied! In general, it is advantageous in terms of convergence speed if there is some asymmetry in the distribution of positive and negative eigenvalues of preconditioned indefinite symmetric matrices. However, it is then considerably more complicated to take account of this asymmetry in a MINRES convergence bound (but see Wathen, Fischer and Silvester 1995).

6. Non-symmetric matrices

As already mentioned, all preconditioning for non-symmetric problems is heuristic, since descriptive convergence bounds for GMRES or any of the other applicable non-symmetric Krylov subspace iterative methods do not presently exist (but see Pestana and Wathen (2013*b*)). The most straightforward convergence bound for GMRES for diagonalizable matrices, $A = X\Lambda X^{-1}$, where Λ is a diagonal matrix of eigenvalues, comes directly from (2.3) and the residual minimization property:

$$\begin{aligned} \|r_k\| &= \min_{p\in\Pi_k, p(0)=1} \|p(A)r_0\| \\ &\leq \min_{p\in\Pi_k, p(0)=1} \|p(X\Lambda X^{-1})\|\,\|r_0\| \end{aligned}$$

$$\begin{aligned}
&= \min_{p\in\Pi_k,p(0)=1} \|Xp(\Lambda)X^{-1}\|\,\|r_0\| \\
&\le \|X\|\,\|X^{-1}\| \min_{p\in\Pi_k,p(0)=1} \|p(\Lambda)\|\,\|r_0\| \\
&\le \|X\|\,\|X^{-1}\| \min_{p\in\Pi_k,p(0)=1} \max_{z\in\mathcal{R}} |p(z)|\,\|r_0\|, \qquad (6.1)
\end{aligned}$$

where $\mathcal{R}$ is any region containing the eigenvalues. Since it is sometimes possible to find regions $\mathcal{R}$ which contain all of the eigenvalues, but rarely possible to obtain any reasonable estimates of the condition number $\|X\|\|X^{-1}\|$ of the eigenvector matrices X, very often in the literature, the eigenvalues of $P^{-1}A$ are estimated or bounded as a guide to the quality of a preconditioner. Certainly it is usually true that if the eigenvalues are widely spread through a large region of $\mathbb{C}$ then slow convergence can follow!

There are alternative approaches for obtaining bounds for GMRES convergence; see for example Greenbaum (1997, Section 3.2). One can generate such bounds, for example, with use of the *field of values* or *numerical range*

$$\mathcal{F}(A) = \{\overline{x}^T Ax, x \in \mathbb{C}^n, \overline{x}^T x = 1\},$$

provided $0 \notin \mathcal{F}(A)$. The obtained bounds can be rather pessimistic, but field-of-values convergence bounds have been used, for example, to rigorously establish the convergence of GMRES in a number of iterations independent of the mesh size, h, for certain preconditioned PDE problems; see for example Klawonn and Starke (1999), Loghin and Wathen (2004) and Benzi and Olshanskii (2011). GMRES convergence bounds can be based on other sets in the complex plane: ϵ-pseudospectral sets, for example, have been used to analyse initial stagnation of (unpreconditioned) GMRES for convection–diffusion problems (Trefethen and Embree 2005, Chapter 26).

One aspect which arises in the non-symmetric but not the symmetric case is the possibility to left-precondition

$$P^{-1}Ax = P^{-1}b,$$

to right-precondition

$$AP^{-1}y = b, \quad x = P^{-1}y,$$

or, if P is available in split form $P = P_1P_2$, to use

$$P_1^{-1}AP_2^{-1}y = P_1^{-1}b, \quad x = P_2^{-1}y.$$

In every case the coefficient matrix is generally again non-symmetric, and obvious similarity transformations involving P or the P_i show that all three coefficient matrices are mathematically similar and so have the same eigenvalues. There is little evidence, however, that any one of these is better than the others, in terms of its effect on the convergence rate of a Krylov subspace method, even though, theoretically, this could be the case. Traditionally, right-preconditioning is favoured since then GMRES, for example, minimizes the residual $r_k = b - Ax_k$ for the original linear system; left-preconditioning would lead to minimization of the preconditioned residual $P^{-1}r_k$. It is certainly not always clear which is preferable: a good preconditioner fortunately seems to give fast convergence for all three forms in practice.

In the context of PDEs, convection–diffusion problems present significant challenges for stable discretization – without excessive numerical diffusion or non-physical oscillations – as well as for preconditioning and iterative solvers. A key aspect is simply equation ordering. Even such a simple method as stationary Gauss–Seidel iteration can be good for such problems if the variables are ordered in the direction of the convection, but it can be very poor otherwise; see Elman and Chernesky (1993) or Elman *et al.* (2014*b*, Chapter 7).

We mention also that there is a 'flexible' version of GMRES which allows for a preconditioner that varies between iterations (Saad 1993). This can be useful, for example, when employing another iteration as a preconditioner, but this flexible GMRES method does not necessarily work as well as regular GMRES. It can sometimes be better to just use a fixed number of inner iterations so that the preconditioner is a fixed operator for every iteration; if such inner iterative steps represent a fixed linear process, then at least the limited guarantees that come with GMRES then remain in place.

6.1. Incomplete factorization

Just as a Cholesky factorization for a symmetric and positive definite matrix can be incompletely formed, an LU factorization (equivalent to Gauss elimination) for a non-symmetric matrix can be performed incompletely by prespecifying allowable fill or thresholding or both. Perhaps such ILU preconditioning methods and their variants are the most widely used for non-symmetric problems, and software is widely available.

Because the product of triangular matrix factors is computed for ILU, for problems with an implied directionality such as convection–diffusion, ordering of the variables can lead to one of the factors being less important (and perhaps more nearly diagonal).

Although, in general, there is little we can say about ILU preconditioning for non-symmetric matrices, GMRES with ILU is a 'go-to' technique for very many practical problems, and therefore much largely empirical research continues to explore every aspect of the various possibilities, for example, multilevel variants, block variants, and parallel implementation issues. We again refer to Benzi (2002), as the huge literature is fairly evenly cited. Because of its significance, particularly in the important area of petroleum reservoir modelling, we specifically mention the nested ILU methods originally derived by Appleyard and Cheshire (1983), which use partial eliminations and are designed to take advantage of parallel computation; related ideas were developed, for example, by Saad (1996). See also Vassilevski (2008) for a comprehensive treatment of these methods in a more general framework.

6.2. Multigrid

There is no inherent obstacle to applying multigrid or multilevel ideas for non-symmetric problems. However, smoothing is generally not so easy and requires more work than for symmetric problems, and care must be taken so that coarse grid correction does not reintroduce high frequencies. Without attention to these aspects, the methods simply do not work at all well. Since such issues vary with differing applications, we briefly discuss only the important problem associated with the convection–diffusion equation.

Smoothing – or perhaps more appropriately in this setting, 'sweeping' – should ideally be something like a Gauss–Seidel iteration with the variables ordered in the direction of the convection. For problems with recirculation, one can try multidirectional sweeping, so that all parts of the flow have at least one of the directional sweeps that is approximately in the convection direction. Coarse grid correction then requires care (in grid transfers and/or coarse grid operator). In the convection–diffusion geometric multigrid method due to Ramage (1999), for example, the coarse grid operator is recomputed on each coarse grid with – crucially – streamline upwinding appropriate to that grid size. The point is that when a grid under-resolves any layer-like phenomenon which arises with convection – this must happen at some level if one has a true multigrid concept – then oscillatory solutions are commonly the result. The reintroduction of such high frequencies would require more significant sweeping or smoothing, and it is preferable to provide stabilization appropriate to each level of grid so that smooth coarse grid corrections result.

Multigrid for hyperbolic problems is thought about rather differently than for elliptic problems, and the idea of directional sweeping, rather than smoothing, is key. The coarse grids allow one to sweep errors out of the boundary of the domain more rapidly than using just a single grid. Again, transport is the main issue, not smoothing.

Algebraic multigrid ideas have been tried and can work well for some problems. The computation of 'coarse grids' (or in reality, spaces of coarse variables) by purely algebraic means can identify layer phenomena, for example, and automatically semi-coarsen. In particular, widely used coarsening algorithms seem to detect anisotropy and lead to appropriate coarse spaces/grids. All is not rosy, however, and AMG software can quite simply fail. There is a particular danger that simply does not arise with geometric multigrid ideas, namely that the coarse grid can have a significant proportion of the variables of the fine grid; that is, coarsening does not lead to a sufficiently significant reduction in the number of variables. If this occurs, then many levels are required and efficiency can be severely compromised. Active research continues on AMG for non-symmetric problems; see for example Notay (2012).

Multigrid methodology is generally applied as a preconditioner for non-symmetric problems.

6.3. Sparse approximate inverses

Norm minimization readily leads to sparse approximate inverses similarly to the symmetric case. A biconjugation based method (AINV) and its more robust stabilized form (SAINV) are, however, perhaps the most widely considered of such methods in the non-symmetric case (Benzi and Tuma 1998, Benzi *et al.* 2000).

6.4. Symmetric/skew splitting

Every non-symmetric real matrix can be uniquely split into its symmetric and skew-symmetric parts:

$$A = \underbrace{\tfrac{1}{2}(A + A^T)}_{H} + \underbrace{\tfrac{1}{2}(A - A^T)}_{S}.$$

(The H stands for Hermitian, since the same idea obviously applies in the complex case when transpose is replaced by conjugate transpose.) When H is positive definite, preconditioning of A by H gives a shifted skew-symmetric matrix for which a special three-term recurrence Krylov subspace method exists (Concus and Golub 1976, Widlund 1978), and convergence depends on the size of $1 + |\lambda_{\max}(H^{-1}S)|$ (Widlund 1978, Freund and Ruscheweyh 1986). Although this approach has been considered for some applications – see for example Elman and Schultz (1986) in the context of convection–diffusion problems – the so-called HSS (Hermitian skew-Hermitian splitting) methods originally introduced by Bai, Golub and Ng (2003) have been rather more widely investigated. They proved the convergence of an alternate iteration using shifted Hermitian and shifted skew-Hermitian parts of the form

$$\begin{aligned}(\alpha I + H)x^{(k+1/2)} &= (\alpha I - S)x^{(k)} + b,\\ (\alpha I + S)x^{(k+1)} &= (\alpha I - H)x^{(k+1/2)} + b\end{aligned}$$

for $Ax = b$. Though convergence is guaranteed independently of the choice of the parameter $\alpha > 0$, it can be slow. Nevertheless, many variants of using this idea as a preconditioner have been suggested, as have related structural splitting ideas, and good results are reported for some; see for example Benzi, Ng, Niu and Wang (2011*a*).

6.5. Non-symmetric saddle point systems

Saddle point problems must essentially be symmetric as in Section 5.2, but matrices of the structure (5.5) arise in which the block A is non-symmetric. A key example is provided by linearizations of the incompressible Navier–Stokes equations; see Elman *et al.* (2014*b*, Chapters 8 and 9). It is these problems that we refer to as non-symmetric saddle point problems.

For a non-symmetric saddle point matrix, the results of Murphy *et al.* (2000) still hold, and block diagonal preconditioners can still be applied. However, since there is no symmetry to lose, the block triangular preconditioners

$$P = \begin{bmatrix} \widehat{A} & 0 \\ B & \pm\widehat{S} \end{bmatrix} \quad \text{or} \quad P = \begin{bmatrix} \widehat{A} & B^T \\ 0 & \pm\widehat{S} \end{bmatrix}$$

will give in the 'ideal' case $\widehat{A} = A, \widehat{S} = S = BA^{-1}B^T$, a preconditioned system with either just the two distinct eigenvalues ± 1 or even just the single eigenvalue of 1, depending on whether the $+$ or the $-$ sign is selected, respectively (Murphy *et al.* 2000). Even with the single eigenvalue 1, P is not the inverse, so something has to give! In fact it is diagonalizability that is lost in selecting the $-$ sign; the minimum polynomial is then $(\lambda - 1)^2$, so two iterations would be required, exactly as for the $+$ case. However, in the practical case when the approximations $\widehat{A}, \widehat{S}$ are not ideal, it is usually the $-$ sign which is chosen, so that all eigenvalues are clustered around 1, rather than the two clusters around ± 1 that arise with the $+$ choice, though it is not clear that this is to be preferred (Ymbert, Embree and Sifuentes 2015).

Approximations of the Schur complement remain a key aspect of any such block preconditioning approach. For the common Oseen linearization of the incompressible Navier–Stokes problem, the matrix block A comes from a vector convection–diffusion operator, so that an appropriate multigrid approach, as outlined above, or any other such good approximation of (preconditioner for) convection–diffusion, is required as $\widehat{A}$. To identify an approximate Schur complement requires more in-depth consideration, but there are now two excellent candidates: the *pressure convection–diffusion* (PCD) (Kay, Loghin and Wathen 2002) and the *least-squares commutator* (LSC) preconditioners. Both are derived, described and analysed in Elman *et al.* (2014*b*, Chapter 9). The PCD method is in some ways simpler, but it requires the construction of a convection–diffusion operator on the pressure space (the space of Lagrange multipliers), which is not required for the problem formulation itself. A purely algebraic way to compute this operator based on sparse approximate inverse technology was, however, described by Elman *et al.* (2006). The LSC method is derived via a commutator argument and is defined purely in terms of the matrix blocks which arise naturally for the Oseen problem. For a comparison and full consideration of advantages and disadvantages of these two approaches, see Elman *et al.* (2014*b*, Chapter 9).

The augmentation or augmented Lagrangian approach has also been applied to non-symmetric saddle point problems. Although some concerns have been expressed regarding possible distortion of the norm in which convergence occurs, good results have been reported even for large Reynolds number (Benzi, Olshanskii and Wang 2011*b*).

Earlier approaches for this problem involved segregated treatment of pressure and velocity and predate the preconditioned Krylov subspace iteration technology. It is perhaps worth commenting here that this example is one of the first for which block preconditioners comprising blocks designed for different sets of variables allow treatment of different 'physics' within the preconditioner, whilst GMRES or some other Krylov subspace method is applied to the complete coupled problem. This is an important paradigm.

Such is its importance that it is in the context of non-symmetric saddle point systems, arising from approximation of the Oseen problem, that perhaps the most refined analysis of preconditioned GMRES convergence has been pursued. There are several eigenvalue analyses for different preconditioners (see, *e.g.*, Elman and Silvester 1996), although these do not lead to rigorous estimates as explained above. It is in the context of this problem, however, that field-of-values convergence analysis has been successfully employed to rigorously bound preconditioned GMRES convergence and to elucidate its dependence on or independence of the mesh size, h, and of the Reynolds number; see for example the analysis of Loghin and Wathen (2004) for PCD, and Benzi and Olshanskii (2011) for block triangular preconditioning of an augmented Lagrangian formulation.

6.6. Circulant and semi-circulant preconditioning

Our opening example already indicates the potential for circulant preconditioning, at least for symmetric Toeplitz matrices. In fact the applicability of such ideas is rather wider, though the theoretical justification is less complete for non-symmetric problems.

The analysis for symmetric and positive definite Toeplitz matrices is well described in the books by Chan and Jin (2007) and Ng (2004), who also indicate how similar ideas follow for block Toeplitz matrices with Toeplitz blocks. Indeed, the use of circulant preconditioners for non-symmetric Toeplitz matrices does not really depend on symmetry. However, unless the normal equations are used, convergence guarantees are not readily available, simply because of the general lack of descriptive convergence bounds for non-symmetric matrices. Good computational results have been observed in many situations, and in fact even proved for certain problems via bounding of the eigenvector condition number as well as the eigenvalues and using (6.1) (Otto 1996). A recent observation of Pestana and Wathen (2015*a*), that guaranteed convergence bounds can be achieved for MINRES, for circulant preconditioned non-symmetric Toeplitz matrices via a variable reordering, is also relevant.

7. Some comments on symmetry

We have emphasized the importance of matrix symmetry in this review, in particular that it is rarely sensible to throw away such a property by preconditioning. In the reverse direction, it would be highly advantageous for non-symmetric matrices if it were easily possible to precondition to obtain symmetry of the preconditioned problem without recourse to the normal equations. This possibility seems unlikely without the computation of expensive matrix factorizations. However, there are situations in which a broader consideration of self-adjointness allows many of the attractive properties of symmetric matrices, such as the use of the symmetric Krylov subspace iterative methods CG and MINRES.

A symmetric matrix, $A = A^T$, is simply one which is self-adjoint in the usual Euclidean inner product, that is,

$$\langle Au, v\rangle = v^T Au = u^T Av = \langle Av, u\rangle = \langle u, Av\rangle.$$

In fact, given any inner product

$$\langle u, v\rangle_H = v^T Hu \tag{7.1}$$

defined by a symmetric and positive definite matrix H (all inner products on $\mathbb{R}^n$ arise in this way), the distinction between symmetric and non-symmetric matrices is really the distinction between matrices which are self-adjoint and those which are non-self-adjoint with respect to the inner product. Indeed, in the standard situation when A and P are both symmetric and P is positive definite, then the left-preconditioned matrix $P^{-1}A$ is self-adjoint in $\langle u, v\rangle_P$, since

$$\langle P^{-1}Au, v\rangle_P = v^T PP^{-1}Au = v^T Au = v^T A^T P^{-T} Pu = \langle u, P^{-1}Av\rangle_P.$$

This description of symmetric preconditioning for symmetric matrices, without destroying self-adjointness, provides an alternative to the approach in Section 3. The CG method, for example, can be employed for a general inner product (7.1) for any matrix that is self-adjoint in this inner product and satisfies the positive definiteness condition $\langle Au, u\rangle_H > 0$ for all $u \in \mathbb{R}^n - \{0\}$.

In general, any matrix A is self-adjoint in (7.1) if and only if, for all $u, v \in \mathbb{R}^n$,

$$v^T HAu = \langle Au, v\rangle_H = \langle u, Av\rangle_H = v^T A^T Hu, \tag{7.2}$$

which is the same condition as $HA = A^T H$, so we must have $A^T = HAH^{-1}$ for some positive definite symmetric matrix H. It can be shown that such a matrix H exists if and only if the real matrix A is diagonalizable and has real eigenvalues (Taussky 1972). Thus the use of self-adjointness in non-standard inner products is limited, but interesting examples do exist where, via left- or right-preconditioning, non-symmetric preconditioned matrices arise which are nevertheless self-adjoint in some convenient non-standard inner product; see Stoll and Wathen (2008). Since the inner product must be used for computation of scalar products (dot products) arising in Krylov subspace methods, there is additional arithmetic overhead in the use of $H \neq I$ unless other structures mitigate this.

We present just one example – perhaps the most well-known – of the use of non-standard inner products; exceptionally, it allows the robust application of CG in a non-standard inner product for an indefinite symmetric saddle point problem by creating a self-adjoint and positive definite matrix in that inner product (Bramble and Pasciak 1988). For other examples, see Klawonn (1998) and Benzi and Simoncini (2006), and see Stoll and Wathen (2008) and Pestana and Wathen (2013*a*) for how examples of this structure can be combined to yield further cases of possible utility (so-called *combination preconditioning*).

In the Bramble and Pasciak method, one preconditions the symmetric and indefinite saddle point matrix as given in (5.5) on the left with the block lower triangular matrix

$$P = \begin{bmatrix} \widehat{A} & 0 \\ B & -I \end{bmatrix},$$

resulting in the non-symmetric matrix

$$P^{-1}\mathcal{A} = \begin{bmatrix} \widehat{A}^{-1}A & \widehat{A}^{-1}B^T \\ B\widehat{A}^{-1}A - B & B\widehat{A}^{-1}B^T \end{bmatrix}.$$

This turns out to be self-adjoint in the inner product (7.1) (*i.e.*, it satisfies the condition (7.2)), where

$$H = \begin{bmatrix} A - \widehat{A} & 0 \\ 0 & I \end{bmatrix}.$$

Moreover, $P^{-1}\mathcal{A}$ is positive definite in this inner product. Note that it is necessary for $A - \widehat{A}$ to be positive definite in order to have an inner product (theoretically – and quite often in practice – this can be achieved by scaling of $\widehat{A}$), and in fact the positive definiteness also requires this condition.

Perhaps this appears to be a rather exotic structure, but much practical use has been made of this particular example; the existence of other really useful examples remains largely unrealized but rather intriguing.

A final comment concerns the possibility of non-symmetric preconditioning of symmetric systems. We have earlier noted that such preconditioning would destroy symmetry in general and is thus undesirable. However, in the conceivable, but presumably rare, situation that $A = A^T, P \neq P^T$ but $P^{-1}A = AP^{-T}$ – that is, P^{-T} is one of the matrices that is self-adjoint in the (possibly indefinite) inner product related to the symmetric matrix A – then it would be possible to use CG for $P^{-1}Ax = P^{-1}b$ when $P^{-1}A$ is positive definite or MINRES if $P^{-1}A$ is indefinite. The CG error would converge in $\|\cdot\|_{P^{-1}A}$, and for MINRES it would be $P^{-1}(b - Ax)$, which would converge in $\|\cdot\|_I$. We do not know of any instance of this structure for non-symmetric P, but the symmetric preconditioner $P^{-1} = q(A)$ for any polynomial, q, clearly does commute with A, and there is some literature on such polynomial preconditioning (see for example O'Leary 1991).

8. Some comments on practical computing

Some important general ideas remain to be mentioned that do not in themselves comprise a particular preconditioning approach, but which have broad significance.

First we mention deflation. It can be helpful – sometimes extremely helpful – to remove certain subspaces from a Krylov subspace. For example, if one has a symmetric and positive definite matrix with a reasonable spread of eigenvalues, except for a few very small eigenvalues for which one has some knowledge of the corresponding eigenspace, then CG convergence could be much improved if the effect of this eigenspace were removed or ignored. This is the idea of deflation: if the eigenspace spanned by the eigenvectors corresponding to the m smallest eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ can be deflated, then the *effective condition number* is $\kappa = \lambda_{\max}/\lambda_{m+1}$, and it is this which appears in the CG convergence bound (3.1). The crucial step is to identify the deflating subspace.

The deflation idea is most effective when the problem naturally presents such a space. For example, see Brandt and Ta'asan (1986), who deflate the low-frequency eigenspaces which correspond to negative eigenvalues for a Helmholtz problem, and Jonsthovel *et al.* (2012), who describe a prototypical application involving composite materials. However, various subspaces which arise from iterations of a previous linear system solve, or other sources, can be used. In particular, when restarting GMRES, one can use information from previous iterates via deflation. There are many papers describing various approaches: for symmetric and positive definite matrices, see for example Frank and Vuik (2001), and for non-symmetric matrices, see for example Erhel, Burrage and Pohl (1995), Sifuentes, Embree and Morgan (2013) and Giraud, Gratton, Pinel and Vasseur (2010). When solving sequences of linear systems, the related idea of 'recycling' can be considered (Parks *et al.* 2006). Deflation is central to the iterative computation of eigenvalues (Lehoucq and Sorensen 1996).

As parallel computing becomes more widespread, preconditioners which are parallelizable or inherently parallel become more important. After all, it is large-scale computation for which preconditioned iterative solution technology is generally being developed. Very simple preconditioning such as diagonal scaling is obviously parallel, but the utility of such an approach is limited, as we have described. Methods based on incomplete factorization are more sequential in design, and although there are parallel variants, often speed of convergence is compromised by ordering and separating variables, *etc.*, to obtain the necessary independence of calculations. Such is the importance of multigrid and AMG methods that parallel forms retaining much of the potency of sequential implementation are critical. There are, however, compromises to be made in going parallel (Worley 1991). There is much to be considered here, but this is not an article on parallel multigrid. The paradigm of domain decomposition is clearly ideal for parallel computation and this is precisely what it was invented for. Sparse approximate inverses were precisely developed because they appeared to be a tractable way to compute a purely algebraic ('given a matrix') preconditioner in parallel. Their effectiveness, however, remains unclear. One beneficial use of these ideas appears to be in contexts where one is not seeking to approximate an inverse, but rather seeking some more limited goal like minimizing a commutator; see for example Elman *et al.* (2006).

Since preconditioning is used with iterative methods, sensible criteria for stopping an iteration must be chosen. In the 'given a matrix' context, some fairly arbitrary small tolerance for a residual norm is usually specified. When one is 'given a problem', however, there can be criteria appropriate to the problem. In particular for PDE problems, reducing iteration error to orders of magnitude less than discretization error is untenable and rather wasteful; practical criteria are given by Arioli (2004) and Arioli, Loghin and Wathen (2005). An important point is that a larger-dimensional matrix for a given problem comes from a more accurate discretization, so any stopping tolerance should also in general be smaller on finer grids; see Elman *et al.* (2014*b*, Section 2.1.2).

In the general context of solution of linear systems of equations, it is often regarded as desirable to reduce the number of variables/degrees of freedom if this is easy; such thinking is associated completely with direct solution methods. In the context of iterative methods, it can be preferable to have a larger-dimensional problem if it is then easier to identify and apply an appropriate preconditioner. That is, it can be convenient even to enlarge the dimension of given systems if the resulting matrices are more readily preconditioned, as long as the storage requirement for the enlarged matrix and the associated vectors does not become unacceptable. In finite element analysis, for example, the idea of 'static condensation' is precisely the elimination of variables with only local connections, but although it slightly reduces the number of free variables, and hence the matrix dimension, the resulting smaller matrices can be less easy to precondition, even though they often possess a slightly reduced condition number. More radical is the idea of the *first-order system least-squares* (FOSLS) approach, where extra dependent variables are explicitly introduced, making a rather larger-dimensional system in general, but a system for which multigrid solvers can be readily applied. For an application of this idea, see de Sterck, Manteuffel, McCormick and Olson (2004), for example.

A further example of the freedom and possibilities is the 'all-at-once', 'one-shot' or 'monolithic' approach, where preconditioning for separate parts of a problem – perhaps separate physical processes – are employed together with a Krylov subspace method for the whole of a coupled problem for which the required matrix–vector products are computable in a matrix-free way, that is, without explicit storage and operation on the complete two-dimensional array. See, for example, Heil, Hazel and Boyle (2008), Haber and Ascher (2001) or Rees, Stoll and Wathen (2010*b*). When matrix–vector products are readily computed, there is always a Krylov subspace method that can be applied to the monolithic problem; the representation of the component parts can very conveniently be the job of a block-structured preconditioner. Good examples of this kind of approach are recent ideas for preconditioning problems coming from magnetohydrodynamics (Phillips *et al.* 2014, Wathen 2014) and from geodynamics (Rhebergen, Wells, Katz and Wathen 2014).

A final comment is aimed at anyone who would like to experiment and see the potential of preconditioning in an easy computational setting. The freely downloadable IFISS software (Elman, Ramage and Silvester 2007, Elman, Ramage and Silvester 2014*a*), which runs under basic MATLAB or OCTAVE, enables such experiments.

9. Summary

We conclude with a few general guidelines on preconditioner design and selection.

- Keep to the original structure when preconditioning (*e.g.*, preserve symmetry, partitioning into blocks, *etc.*)

- The more knowledge about a problem that can be represented in a preconditioner, the better.

- The work in applying the preconditioner should ideally be commensurate with the work in matrix–vector multiplication: that would be $\mathcal{O}(n)$ for a sparse matrix and $\mathcal{O}(n^2)$ for a general full matrix.

- It may be easier to precondition a matrix of larger dimension, for which structures are clearer, than to eliminate variables, giving a less structured matrix of smaller dimension.

There is, of course, no such concept as a best preconditioner: the only two candidates for this would be $P = I$, for which the preconditioning takes no time at all, and $P = A$, for which only one iteration would be required for solution by any iterative method. However, every practitioner knows when they have a good preconditioner which enables feasible computation and solution of problems. In this sense, preconditioning will always be an art rather than a science.

Acknowledgements

I am grateful to Michele Benzi, Howard Elman and David Silvester for reading an early draft. Their comments and suggestions have led to significant improvements.

Parts of this survey were written whilst visiting CERFACS, Toulouse, France and the Memorial University of Newfoundland, Canada. I am grateful to Serge Gratton, Xavier Vasseur, Hermann Brunner and Scott MacLachlan for their support and encouragement.

REFERENCES[1]

P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent and J. Koster (2000), 'MUMPS: A general purpose distributed memory sparse solver'. In *Proc. PARA2000: 5th International Workshop on Applied Parallel Computing*, Springer, pp. 122–131.

J. R. Appleyard and I. M. Cheshire (1983), Nested factorization. Society of Petroleum Engineers, SPE 12264, presented at the Seventh SPE Symposium on Reservoir Simulation, San Francisco, 1983.

M. Arioli (2004), 'A stopping criterion for the conjugate gradient algorithm in a finite element method framework', *Numer. Math.* **97**, 1–24.

M. Arioli, D. Loghin and A. J. Wathen (2005), 'Stopping criteria for iterations in finite element methods', *Numer. Math.* **99**, 381–410.

D. N. Arnold, R. S. Falk and R. Winther (1997), 'Preconditioning in $H(\mathrm{div})$ and applications', *Math. Comp.* **66**, 957–984.

Z.-Z. Bai, G. H. Golub and M. K. Ng (2003), 'Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems', *SIAM J. Matrix Anal. Appl.* **24**, 603–626.

N. S. Bakhvalov (1966), 'On the convergence of a relaxation method with natural constraints on the elliptic operator', *USSR Comp. Math. and Math. Phys.* **6**, 101–135.

[1] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

M. Bebendorf (2008), *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Vol. 63, Lecture Notes in Computational Science and Engineering, Springer.

M. Benzi (2002), 'Preconditioning techniques for large linear systems: A survey', *J. Comput. Phys.* **182**, 418–477.

M. Benzi and M. A. Olshanskii (2011), 'Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem', *SIAM J. Numer. Anal.* **49**, 770–788.

M. Benzi and V. Simoncini (2006), 'On the eigenvalues of a class of saddle point matrices', *Numer. Math.* **103**, 173–196.

M. Benzi and M. Tuma (1998), 'A sparse approximate inverse preconditioner for nonsymmetric linear systems', *SIAM J. Sci. Comput.* **19**, 968–994.

M. Benzi and M. Tuma (2003), 'A robust incomplete factorization preconditioner for positive definite matrices', *Numer. Linear Algebra Appl.* **10**, 385–400.

M. Benzi and A. J. Wathen (2008), Some preconditioning techniques for saddle point problems. In *Model Order Reduction: Theory, Research Aspects and Applications* (W. H. Schilders, H. A. van der Vorst and J. Rommes, eds), Springer, pp. 195–211.

M. Benzi, J. K. Cullum and M. Tuma (2000), 'Robust approximate inverse preconditioning for the conjugate gradient method', *SIAM J. Sci. Comput.* **22**, 1318–1332.

M. Benzi, G. H. Golub and J. Liesen (2005), Numerical solution of saddle point problems. In *Acta Numerica*, Vol. 14, Cambridge University Press, pp. 1–137.

M. Benzi, C. D. Meyer and M. Tuma (1996), 'A sparse approximate inverse preconditioner for the conjugate gradient method', *SIAM J. Sci. Comput.* **17**, 1135–1149.

M. Benzi, M. Ng, Q. Niu and Z. Wang (2011*a*), 'A relaxed dimensional factorization preconditioner for the incompressible Navier–Stokes equations', *J. Comput. Phys.* **230**, 6185–6202.

M. Benzi, M. A. Olshanskii and Z. Wang (2011*b*), 'Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations', *Internat. J. Numer. Meth. Fluids* **66**, 486–508.

L. Bergamaschi, J. Gondzio and G. Zilli (2004), 'Preconditioning indefinite systems in interior point methods for optimization', *Comput. Optim. Appl.* **28**, 149–171.

M. Bern, J. R. Gilbert, B. Hendrickson, N. Nguyen and S. Toledo (2006), 'Support-graph preconditioners', *SIAM J. Matrix Anal. Appl.* **27**, 930–951.

A. Björck (1996), *Numerical Methods for Least Squares Problems*, SIAM.

J. W. Boyle, M. D. Mihajlovic and J. A. Scott (2010), 'HSL_MI20: An efficient AMG preconditioner for finite element problems in 3D', *Internat. J. Numer. Meth. Engng* **82**, 64–98.

D. Braess and P. Peisker (1986), 'On the numerical solution of the biharmonic equation and the role of squaring matrices for preconditioning', *IMA J. Numer. Anal.* **6**, 393–404.

J. Bramble and J. Pasciak (1988), 'A preconditioning technique for indefinite system resulting from mixed approximations of elliptic problems', *Math. Comp.* **50**, 1–17.

J. H. Bramble, J. E. Pasciak and J. Xu (1990), ‘Parallel multilevel preconditioners’, *Math. Comp.* **55**, 1–22.

A. Brandt (1977), ‘Multilevel adaptive solutions to boundary-value problems’, *Math. Comp.* **31**, 333–390.

A. Brandt and S. Ta’asan (1986), Multigrid methods for nearly singular and slightly indefinite problems. In *Multigrid Methods II* (W. Hackbusch and U. Trottenberg, eds), Vol. 1228 of *Lecture Notes in Mathematics*, Springer, pp. 99–121.

O. Bröker, M. J. Grote, C. Mayer and A. Reusken (2001), ‘Robust parallel smoothing for multigrid via sparse approximate inverses’, *SIAM J. Sci Comput.* **23**, 1395–1416.

R. H.-F. Chan and X.-Q. Jin (2007), *An Introduction to Iterative Toeplitz Solvers*, SIAM.

T. F. Chan and T. P. Mathew (1994), Domain decomposition algorithms. In *Acta Numerica*, Vol. 3, Cambridge University Press, pp. 61–143.

S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A. J. van der Veen and D. White (2005), ‘Some fast algorithms for sequentially semiseparable representation’, *SIAM J. Matrix Anal. Appl.* **27**, 341–364.

K. Chen (2005), *Matrix Preconditioning Techniques and Applications*, Cambridge University Press.

P. Concus and G. H. Golub (1976), A generalized conjugate gradient method for nonsymmetric systems of linear equations. In Vol. 134 of *Lecture Notes in Economics and Mathematical Systems* (R. Glowinski and P. L. Lions eds), Springer, pp. 56–65.

J. W. Cooley and J. W. Tukey (1965), ‘An algorithm for the machine calculation of complex Fourier series’, *Math. Comp.* **19**, 297–301.

T. A. Davis (2004), ‘Algorithm 832: UMFPACK V4.3 an unsymmetric-pattern multifrontal method’, *ACM Trans. Math. Software* **30**, 196–199.

H. de Sterck, T. Manteuffel, S. McCormick and L. Olson (2004), ‘Least-squares finite element methods and algebraic multigrid solvers for linear hyperbolic PDEs’, *SIAM J. Sci Comput.* **26**, 31–54.

H. S. Dollar and A. J. Wathen (2006), ‘Approximate factorization constraint preconditioners for saddle-point matrices’, *SIAM J. Sci. Comput.* **27**, 1555–1572.

H. S. Dollar, N. I. M. Gould, W. H. A. Schilders and A. J. Wathen (2006), ‘Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems’, *SIAM J. Matrix Anal. Appl.* **28**, 170–189.

T. F. Dupont, R. P. Kendall and H. H. Rachford (1968), ‘An approximate factorization procedure for solving self-adjoint elliptic difference equations’, *SIAM J. Numer. Anal.* **5**, 554–573.

H. C. Elman and M. P. Chernesky (1993), ‘Ordering effects on relaxation methods applied to the discrete one-dimensional convection–diffusion equation’, *SIAM J. Numer. Anal.* **30**, 1268–1290.

H. C. Elman and M. H. Schultz (1986), ‘Preconditioning by fast direct methods for non-self-adjoint nonseparable elliptic equations’, *SIAM J. Numer. Anal.* **23**, 44–57.

H. C. Elman and D. J. Silvester (1996), ‘Fast nonsymmetric iterations and preconditioning for the Navier–Stokes equations’, *SIAM J. Sci. Comput.* **17**, 33–46.

H. C. Elman and X. Zhang (1995), 'Algebraic analysis of the hierarchical basis preconditioner', *SIAM J. Matrix Anal. Appl.* **16**, 192–206.

H. C. Elman, O. G. Ernst and D. P. O'Leary (2001), 'A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations', *SIAM J. Sci. Comput.* **23**, 1291–1315.

H. C. Elman, V. E. Howle, J. N. Shadid, R. Shuttleworth and R. Tuminaro (2006), 'Block preconditioners based on approximate commutators', *SIAM J. Sci. Comput.* **27**, 1651–1668.

H. C. Elman, A. Ramage and D. J. Silvester (2007), 'Algorithm 886: IFISS, A Matlab toolbox for modelling incompressible flow', *ACM Trans. Math. Software* **33**, #14.

H. C. Elman, A. Ramage and D. J. Silvester (2014*a*), 'IFISS: A computational laboratory for investigating incompressible flow problems', *SIAM Rev.* **56**, 261–273.

H. C. Elman, D. J. Silvester and A. J. Wathen (2014*b*), *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, second edition, Oxford University Press.

H. C. Elman, D. J. Silvester and A. J. Wathen (2005), *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press.

B. Engquist and L. Ying (2011*a*), 'Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation', *Comm. Pure Appl. Math.* **64**, 697–735.

B. Engquist and L. Ying (2011*b*), 'Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers', *Multiscale Model. Simul.* **9**, 686–710.

J. Erhel, K. Burrage and B. Pohl (1995), 'Restarted GMRES preconditioned by deflation', *J. Comput. Appl. Math.* **69**, 303–318.

Y. A. Erlangga, C. W. Oosterlee and C. Vuik (2006), 'A novel multigrid based preconditioner for heterogeneous Helmholtz problems', *SIAM J. Sci. Comput.* **27**, 1471–1492.

O. G. Ernst and M. J. Gander (2012), Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical Analysis of Multiscale Problems* (I. G. Graham, T. Y. Hou, O. Lakkis and R. Scheichl, eds), Springer, pp. 325–363.

V. Faber and T. Manteuffel (1984), 'Necessary and sufficient conditions for the existence of a conjugate gradient method', *SIAM J. Numer. Anal.* **21**, 315–339.

R. P. Fedorenko (1964), 'The speed of convergence of one iterative process', *USSR Comp. Math. and Math. Phys.* **4**, 227–235.

B. Fischer (2011), *Polynomial Based Iteration Methods for Symmetric Linear Systems*, SIAM.

B. Fischer, A. Ramage, D. J. Silvester and A. J. Wathen (1998), 'Minimum residual methods for augmented systems', *BIT* **38**, 527–543.

J. Frank and C. Vuik (2001), 'On the construction of deflation-based preconditioners', *SIAM J. Sci. Comput.* **23**, 442–462.

R. W. Freund and N. M. Nachtigal (1991), 'QMR: A quasi-minimal residual method for non-Hermitian linear systems', *Numer. Math.* **60**, 315–339.

R. W. Freund and N. M. Nachtigal (1994), A new Krylov-subspace method for symmetric indefinite linear systems. In *Proc. 14th IMACS World Congress on Computational and Applied Mathematics* (W. F. Ames, ed.), IMACS, pp. 1253–1256.

R. W. Freund and S. Ruscheweyh (1986), 'On a class of Chebyshev approximation problems which arise in connection with a conjugate gradient type method', *Numer. Math.* **48**, 525–542.

M. J. Gander, F. Magoulès and F. Nataf (2002), 'Optimized Schwarz methods without overlap for the Helmholtz equation', *SIAM J. Sci. Comput.* **24**, 38–60.

M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro and M. G. Sala (2006), ML 5.0 Smoothed Aggregation User's Guide. Report SAND2006-2649, Sandia National Laboratories.

M. B. Giles (2008), 'Multilevel Monte Carlo path simulation', *Oper. Res.* **56**, 607–617.

L. Giraud, S. Gratton, X. Pinel and X. Vasseur (2010), 'Flexible GMRES with deflated restarting', *SIAM J. Sci. Comput.* **32**, 1858–1878.

N. I. M. Gould, M. E. Hribar and J. Nocedal (2001), 'On the solution of equality constrained quadratic programming problems arising in optimization', *SIAM J. Sci. Comput.* **23**, 1376–1395.

N. I. M. Gould and V. Simoncini (2009), 'Spectral analysis of saddle point matrices with indefinite leading blocks', *SIAM J. Matrix Anal. Appl.* **31**, 1152–1171.

I. G. Graham and M. J. Hagger (1999), 'Unstructured additive Schwarz-conjugate gradient method for elliptic problems with highly discontinuous coefficients', *SIAM J. Sci. Comput.* **20**, 2041–2066.

L. Grasedyck and W. Hackbusch (2003), 'Construction and arithmetics of $\mathcal{H}$-matrices', *Computing* **70**, 295–334.

A. Greenbaum (1997), *Iterative Methods for Solving Linear Systems*, SIAM.

L. Greengard and V. Rokhlin (1987), 'A fast algorithm for particle simulations', *J. Comput. Phys.* **73**, 325–348.

C. Greif and D. Schötzau (2006), 'Preconditioners for saddle point linear systems with highly singular (1,1) blocks', *Elect. Trans. Numer. Anal.* **22**, 114–121.

C. Greif and D. Schötzau (2007), 'Preconditioners for the discretized time-harmonic Maxwell equations in mixed form', *Numer. Linear Algebra Appl.* **14**, 281–297.

M. J. Grote and T. Huckle (1997), 'Parallel preconditioning with sparse approximate inverses', *SIAM J. Sci. Comput.* **18**, 83–853.

A. Günnel, R. Herzog and E. Sachs (2014), 'A note on preconditioners and scalar products in Krylov subspace methods for self-adjoint problems in Hilbert space', *Elect. Trans. Numer. Anal.* **41**, 13–20.

I. Gustafsson (1978), 'A class of first order factorization methods', *BIT* **18**, 142–156.

E. Haber and U. M. Ascher (2001), 'Preconditioned all-at-once methods for large, sparse parameter estimation problems', *Inverse Probl.* **17**, 1847–1864.

W. Hackbusch (1994), *Iterative Solution of Large Sparse Systems of Equations*, Springer.

W. Hackbusch (1999), 'A sparse matrix arithmetic based on $\mathcal{H}$-matrices I: Introduction to $\mathcal{H}$-matrices', *Computing* **62**, 89–108.

W. Hackbusch, B. K. Khoromskij and S. Sauter (2000), On $\mathcal{H}^2$-matrices. In *Lectures on Applied Mathematics* (H. Bungartz, R. Hoppe and C. Zenger, eds), Springer, pp. 9–29.

M. Heil, A. L. Hazel and J. Boyle (2008), 'Solvers for large-displacement fluid–structure interaction problems: Segregated versus monolithic approaches', *Comput. Mech.* **43**, 91–101.

V. E. Henson and U. M. Yang (2000), 'BoomerAMG: A parallel algebraic multigrid solver and preconditioner', *Appl. Numer. Math.* **41**, 155–177.

M. R. Hestenes and E. Stiefel (1952), 'Methods of conjugate gradients for solving linear problems', *J. Res. Nat. Bur. Stand.* **49**, 409–436.

R. Hiptmair and J. Xu (2007), 'Nodal auxiliary space preconditioning in H(curl) and H(div) spaces', *SIAM J. Numer. Anal.* **45**, 2483–2509.

R. M. Holland, A. J. Wathen and G. J. Shaw (2005), 'Sparse approximate inverses and target matrices', *SIAM J. Sci. Comput.* **26**, 1000–1011.

HSL Mathematical Software Library (2013), A collection of Fortran codes for large scale scientific computation. www.hsl.rl.ac.uk

International Conference on Domain Decomposition Methods (2014). www.ddm.org/conferences.html

V. John and L. Tobiska (2000), 'Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations', *Internat. J. Numer. Meth. Fluids* **33**, 453–473.

T. B. Jonsthovel, M. B. van Gijzen, S. C. MacLachlan, C. Vuik and A. Scarpas (2012), 'Comparison of the deflated preconditioned conjugate gradient method and algebraic multigrid for composite materials', *Comput. Mech.* **50**, 321–333.

D. Kay, D. Loghin and A. J. Wathen (2002), 'A preconditioner for the steady-state Navier–Stokes equations', *SIAM J. Sci. Comput.* **24**, 237–256.

C. Keller, N. I. M. Gould and A. J. Wathen (2000), 'Constraint preconditioning for indefinite linear systems', *SIAM J. Matrix Anal. Appl.* **21**, 1300–1317.

A. Klawonn (1998), 'Block-triangular preconditioners for saddle point problems with a penalty term', *SIAM J. Sci. Comput.* **19**, 172–184.

A. Klawonn and G. Starke (1999), 'Block triangular preconditioners for nonsymmetric saddle point problems: Field-of-value analysis', *Numer. Math.* **81**, 577–594.

L. Y. Kolotilina and A. Y. Yeremin (1993), 'Factorized sparse approximate inverse preconditioning I: Theory', *SIAM J. Matrix Anal. Appl.* **14**, 45–58.

J. Korzak (1999), 'Eigenvalue relations and conditions of matrices arising in linear programming', *Computing* **62**, 45–54.

A. L. Laird and M. B. Giles (2002), Preconditioned iterative solution of the 2D Helmholtz equation. Report NA-02/12, Oxford University Computer Laboratory.

R. B. Lehoucq and D. C. Sorensen (1996), 'Deflation techniques for an implicitly restarted Arnoldi iteration', *SIAM J. Matrix Anal. Appl.* **17**, 789–821.

X. S. Li (2005), 'An overview of SuperLU: Algorithms, implementation and user interface', *ACM Trans. Math. Software* **31**, 302–325.

J. Liesen and Z. Strakoš (2013), *Krylov Subspace Methods: Principles and Analysis*, Oxford University Press.

O. E. Livne and A. Brandt (2012), 'Lean Algebraic Multigrid (LAMG): Fast graph Laplacian linear solver', *SIAM J. Sci. Comput.* **34**, B499–B522.

D. Loghin and A. J. Wathen (2004), 'Analysis of preconditioners for saddle-point problems', *SIAM J. Sci. Comput.* **25**, 2029–2049.

M. Magolu Monga Made, R. Beauwens and G. Warzée (2000), 'Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix', *Commun. Numer. Methods Engng* **16**, 801–817.

J. Málek and Z. Strakoš (2014), *Preconditioning and the Conjugate Gradient Method in the Context of Solving PDEs*, SIAM.

S. Manservisi (2006), 'Numerical analysis of Vanka-type solvers for steady Stokes and Navier–Stokes flows', *SIAM J. Numer. Anal.* **44**, 2025–2056.

K.-A. Mardal and R. Winther (2011), 'Preconditioning discretizations of systems of partial differential equations', *Numer. Linear Algebra Appl.* **18**, 1–40.

T. P. Mathew (2008), *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*, Springer.

J. A. Meijerink and H. A. van der Vorst (1977), 'An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix', *Math. Comp.* **31**, 148–162.

G. A. Meurant (1999), *Computer Solution of Large Linear Systems*, North-Holland.

M. F. Murphy, G. H. Golub and A. J. Wathen (2000), 'A note on preconditioning for indefinite linear systems', *SIAM J. Sci. Comput.* **21**, 1969–1972.

M. Neytcheva (1999), talk presented at the first *International Conference on Preconditioning*, Minneapolis, 1999.

M. K. Ng (2004), *Iterative Methods for Toeplitz Systems*, Oxford University Press.

Y. Notay (2012), 'Aggregation-based algebraic multigrid for convection–diffusion equations', *SIAM J. Sci. Comput.* **34**, A2288–A2316.

D. P. O'Leary (1991), 'Yet another polynomial preconditioner for the conjugate gradient algorithm', *Linear Algebra Appl.* **154–156**, 377–388.

M. A. Olshanskii and E. E. Tyrtyshnikov (2014), *Iterative Methods for Linear Systems: Theory and Applications*, SIAM.

M. E. G. Ong (1997), 'Hierarchical basis preconditioners in three dimensions', *SIAM J. Sci. Comput.* **18**, 479–498.

K. Otto (1996), 'Analysis of preconditioners for hyperbolic partial differential equations', *SIAM J. Numer. Anal.* **33**, 2131–2165.

C. C. Paige and M. A. Saunders (1975), 'Solution of sparse indefinite systems of linear equations', *SIAM J. Numer. Anal.* **12**, 617–629.

C. C. Paige and M. A. Saunders (1982), 'LSQR: An algorithm for sparse linear equations and sparse least squares', *ACM Trans. Math. Software* **8**, 43–71.

M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson and S. Maiti (2006), 'Recycling Krylov subspaces for sequences of linear systems', *SIAM J. Sci. Comput.* **28**, 1651–1674.

J. Pestana and A. J. Wathen (2013*a*), 'Combination preconditioning of saddle point systems for positive definiteness', *Numer. Linear Algebra Appl.* **20**, 785–808.

J. Pestana and A. J. Wathen (2013*b*), 'On choice of preconditioner for minimum residual methods for non-Hermitian matrices', *J. Comput. Appl. Math.* **249**, 57–68.

J. Pestana and A. J. Wathen (2015*a*), 'A preconditioned MINRES method for nonsymmetric Toeplitz matrices', *SIAM J. Matrix Anal. Appl.*, to appear.

J. Pestana and A. J. Wathen (2015*b*), 'Natural preconditioning and iterative methods for saddle point systems', *SIAM Rev.* **57**, 71–91.

E. G. Phillips, H. C. Elman, E. C. Cyr, J. N. Shadid and R. P. Pawlowski (2014), A block preconditioner for an exact penalty formulation for stationary MHD. Computer Science report CS-TR-5032, University of Maryland.

J. Poulson, B. Engquist, S. Li and L. Ying (2013), 'A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations', *SIAM J. Sci. Comput.* **35**, C194–C212.

A. Quarteroni and A. Valli (1999), *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press.

A. Ramage (1999), 'A multigrid preconditioner for stabilised discretisations of advection–diffusion problems', *J. Comput. Appl. Math.* **101**, 187–203.

T. Rees, H. S. Dollar and A. J. Wathen (2010*a*), 'Optimal solvers for PDE-constrained optimization', *SIAM J. Sci. Comput.* **32**, 271–298.

T. Rees, M. Stoll and A. J. Wathen (2010*b*), 'All-at-once preconditioning in PDE-constrained optimization', *Kybernetika* **46**, 341–360.

S. Rhebergen, G. N. Wells, R. F. Katz and A. J. Wathen (2014), 'Analysis of block-preconditioners for models of coupled magma/mantle dynamics', *SIAM J. Sci. Comput.* **36**, A1960–A1977.

Y. Saad (1993), 'A flexible inner-outer preconditioned GMRES algorithm', *SIAM J. Sci. Comput.* **14**, 461–469.

Y. Saad (1996), 'ILUM: A multi-elimination ILU preconditioner for general sparse matrices', *SIAM J. Sci. Comput.* **17**, 830–847.

Y. Saad (2003), *Iterative Methods for Sparse Linear Systems*, second edition, SIAM.

Y. Saad and M. H. Schultz (1986), 'GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Statist. Comput.* **7**, 856–869.

J. A. Sifuentes, M. Embree and R. B. Morgan (2013), 'GMRES convergence for perturbed coefficient matrices, with application to approximate deflation preconditioning', *SIAM. J. Matrix Anal. Appl.* **34**, 1066–1088.

D. J. Silvester and A. J. Wathen (1994), 'Fast iterative solution of stabilised Stokes systems II: Using general block preconditioners', *SIAM J. Numer Anal.* **31**, 1352–1367.

V. Simoncini and D. B. Szyld (2007), 'Recent computational developments in Krylov subspace methods for linear systems', *Numer. Linear Algebra Appl.* **14**, 1–59.

G. L. G. Sleijpen and D. R. Fokkema (1993), 'BiCGstab(ell) for linear equations involving unsymmetric matrices with complex spectrum', *Elect. Trans. Numer. Anal.* **1**, 11–32.

B. Smith, P. Bjørstat and W. Gropp (1996), *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press.

P. Sonneveld and M. B. van Gijzen (2008), 'IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations', *SIAM J. Sci. Comput.* **31**, 1035–1062.

D. A. Spielman and S.-H. Teng (2014), 'Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems', *SIAM J. Matrix Anal. Appl.* **35**, 835–885.

M. Stoll and A. J. Wathen (2008), 'Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner', *SIAM J. Matrix Anal. Appl.* **30**, 582–608.

G. Strang (1986), 'A proposal for Toeplitz matrix calculations', *Stud. Appl. Math.* **74**, 171–176.

W.-P. Tang and W. L. Wan (2000), 'Sparse approximate inverse smoother for multigrid', *SIAM J. Matrix Anal. Appl.* **21**, 1236–1252.

O. Taussky (1972), 'The role of symmetric matrices in the study of general matrices', *Linear Algebra Appl.* **5**, 147–154.

A. Toselli and O. Widlund (2004), Domain Decomposition Methods: Algorithms and Theory. Vol. 34 of *Springer Series in Computational Mathematics*, Springer.

L. N. Trefethen and M. Embree (2005), *Spectra and Pseudospectra*, Princeton University Press.

U. Trottenberg, C. W. Oosterlee and A. Schüler (2000), *Multigrid*, Academic Press.

P. Tsuji, B. Engquist and L. Ying (2012), 'A sweeping preconditioner for time-harmonic Maxwell's equations with finite elements', *J. Comput. Phys.* **231**, 3770–3783.

M. B. van Gijzen, Y. A. Erlangga and C. Vuik (2007), 'Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian', *SIAM J. Sci. Comput.* **29**, 1942–1958.

H. A. van der Vorst (1992), 'Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems', *SIAM J. Sci. Statist. Comput.* **13**, 631–644.

H. A. van der Vorst (2003), *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press.

C. F. Van Loan (1992), *Computational Frameworks for the Fast Fourier Transform*, SIAM.

P. S. Vassilevski (2008), *Multilevel Block Factorization Preconditioners: Matrix-Based Analysis and Algorithms for Solving Finite Element Equations*, Springer.

R. Verfürth (1984), 'A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem', *IMA J. Numer. Anal.* **4**, 441–455.

A. J. Wathen (1986), 'Realistic eigenvalue bounds for the Galerkin mass matrix', *IMA J. Numer. Anal.* **7**, 449–457.

A. J. Wathen (2007), 'Preconditioning and convergence in the right norm', *Internat. J. Comput. Math.* **84**, 1199–1209.

A. J. Wathen and T. Rees (2009), 'Chebyshev semi-iteration in preconditioning for problems including the mass matrix', *Elect. Trans. Numer. Anal.* **34**, 125–135.

A. J. Wathen, B. Fischer and D. J. Silvester (1995), 'The convergence rate of the minimum residual method for the Stokes problem', *Numer. Math.* **71**, 121–134.

M. P. Wathen (2014), Iterative solution of a mixed finite element discretisation of an incompressible magnetohydrodynamics problem. MSc dissertation, Department of Computer Science, University of British Columbia.

O. Widlund (1978), 'A Lanczos method for a class of nonsymmetric systems of linear equations', *SIAM J. Numer. Anal.* **15**, 801–812.

P. H. Worley (1991), 'Limits on parallelism in the numerical solution of linear partial differential equations', *SIAM J. Sci. Statist. Comput.* **12**, 1–35.

J. Xu (1992), 'Iterative methods by space decomposition and subspace correction', *SIAM Rev.* **34**, 581–613.

G. Ymbert, M. Embree and J. Sifuentes (2015), Approximate Murphy–Golub–Wathen preconditioning for saddle point problems. In preparation.

H. Yserentant (1986), 'On the multi-level splitting of finite element spaces', *Numer. Math.* **49**, 379–412.