SOME PRECONDITIONING TECHNIQUES FOR SADDLE POINT PROBLEMS

MICHELE BENZI* AND ANDREW J. WATHEN†

Abstract. Saddle point problems arise frequently in many applications in science and engineering, including constrained optimization, mixed finite element formulations of partial differential equations, circuit analysis, and so forth. Indeed the formulation of most problems with constraints gives rise to saddle point systems. This paper provides a concise overview of iterative approaches for the solution of such systems which are of particular importance in the context of large scale computation. In particular we describe some of the most useful preconditioning techniques for Krylov subspace solvers applied to saddle point problems, including block and constrained preconditioners.

Key words. sparse linear systems, indefinite matrices, iterative methods, preconditioning

AMS subject classifications. 65F10, 65N22, 65F50.

1. Introduction.

Numerous mathematical models in science and engineering can be stated in the form of constrained minimization problems. Frequently, such problems are infinite-dimensional and highly nonlinear. Discretization results in finite-dimensional problems of large size. These problems are usually replaced by a sequence of quadratic minimization problems subject to linear equality constraints:

$$\min \;\; J(u) = \tfrac{1}{2} u^T A u - f^T u \tag{1.1}$$

$$\text{subject to} \;\; Bu = g \,. \tag{1.2}$$

Here $A \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite, and $B \in \mathbb{R}^{m\times n}$, with $m < n$; $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$ are given vectors. The first-order optimality conditions are given by the linear system

$$\begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} . \tag{1.3}$$

In (1.3), $p \in \mathbb{R}^m$ is a vector of Lagrange multipliers. Linear systems of the form (1.3) are known as *saddle point problems*, since any solution (u, p) of (1.3) is a saddle point of the Lagrangian function

$$\mathcal{L}(u,p) = \tfrac{1}{2} u^T A u - f^T u + (Bu - g)^T p \,.$$

Large linear systems in saddle point form also arise from inherently discrete physical models, such as mechanical structures [40] and RCL circuits [17].

More generally, we consider linear systems of the form

$$\mathcal{A}\, x = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} = b \,, \tag{1.4}$$

with A and B as before and $C \in \mathbb{R}^{m\times m}$ symmetric and positive semidefinite. Systems of the form (1.4) with a nonzero (2,2) block arise, for instance, in the context of interior

*Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (benzi@mathcs.emory.edu). The work of this author was supported in part by the National Science Foundation grant DMS-0511336.

†Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, UK (andy.wathen@comlab.ox.ac.uk).

point methods for constrained optimization [32]. Other examples are provided by mixed finite elements for incompressible flow problems, when some form of pressure stabilization is included in the discretization [13], and by the modeling of slightly compressible material in linear elasticity theory [7].

Typically, $\mathcal{A}$ is large and sparse and (1.4) must be solved iteratively, usually by means of Krylov subspace algorithms [41]. Unfortunately, Krylov methods tend to converge very slowly when applied to saddle point systems, and good preconditioners are needed to achieve rapid convergence. In the last few years, much work has been devoted to developing effective preconditioners for saddle point systems. The goal of this paper is to provide a concise overview of such techniques. Due to space limitations, we focus mainly on three widely applicable classes of preconditioning techniques: block diagonal (or triangular) preconditioners, constraint preconditioners, and HSS preconditioning. For a more extensive survey of these and other techniques, see [3]. See further [13] for a thorough discussion of saddle point problems arising in fluid dynamics.

2. Properties of saddle point systems. If A is nonsingular, the saddle point matrix $\mathcal{A}$ admits the following block triangular factorization:

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} = \begin{bmatrix} I & O \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & O \\ O & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ O & I \end{bmatrix}, \tag{2.1}$$

where $S = -(C + BA^{-1}B^T)$ is the *Schur complement* of A in $\mathcal{A}$. Several important properties of the saddle point matrix $\mathcal{A}$ can be derived on the basis of (2.1). To begin with, it is clear that $\mathcal{A}$ is nonsingular if and only if S is. Furthermore, since (2.1) defines a congruence transformation, we see that $\mathcal{A}$ is indefinite with n positive and m negative eigenvalues if A is symmetric positive definite (SPD).

There are some important applications in which A is symmetric positive semidefinite and singular, in which case there is no block factorization of the form (2.1). If $C = O$ and B has full rank, then $\mathcal{A}$ is invertible if and only if the null spaces of A and B satisfy $\mathcal{N}(A) \cap \mathcal{N}(B) = \{0\}$. In this case $\mathcal{A}$ is, again, indefinite with n positive and m negative eigenvalues. In some important applications A is SPD and B is rank deficient and the linear system (1.4) is singular but consistent. Generally speaking, the singularity of $\mathcal{A}$ does not cause any serious problem for iterative solvers; see [13, Section 5.3] for a discussion.

It is interesting to note that the simple stratagem of changing the sign of the last m equations in (1.4) leads to a linear system with completely different spectral properties. Indeed, assuming that A is SPD and C is symmetric positive semidefinite, it is easy to see that the (nonsymmetric) coefficient matrix

$$\widehat{\mathcal{A}} = \begin{bmatrix} A & B^T \\ -B & C \end{bmatrix} \tag{2.2}$$

is positive definite, in the sense that its spectrum is contained in the right half-plane $\Re(z) > 0$. Hence, $-\widehat{\mathcal{A}}$ is a *stable* matrix, an important property in circuit modeling; see [17, Section 4.3]. Furthermore, when certain (reasonable) conditions on A, B and C are met, it can be shown that $\widehat{\mathcal{A}}$ is diagonalizable and has all the eigenvalues real and positive. In other words, there exists a nonstandard inner product on $\mathbb{R}^{n+m}$ relative to which $\widehat{\mathcal{A}}$ is SPD; see [5] for details.

Regardless of the formulation of the saddle point system (symmetric indefinite or nonsymmetric positive definite), the convergence of Krylov subspace methods is almost always extremely slow unless a good preconditioner is available.

3. Preconditioned Krylov subspace methods. The well-known Conjugate Gradient method [25] which is widely used for the iterative solution of symmetric *definite* matrix systems is not in general robust for indefinite matrix systems. The main iterative approaches for indefinite matrix systems are the MINRES and SYMMLQ algorithms [31] which are based on the Lanczos procedure [28]. These algorithms (see [14] for a comprehensive and accessible description) require any preconditioner to be symmetric and positive definite. An alternative, which allows the use of symmetric and indefinite preconditioning (but has less clear theoretical convergence properties) is the Symmetric QMR (SQMR) method [19]. Even for indefinite problems, however, Conjugate Gradient methods can be employed with specific types of preconditioner: see the section on Constraint Preconditioning below.

The important feature of all of these methods is that at each iteration only one matrix times vector multiplication and a small number of vector operations (dot products and vector updates) are required. For sparse or structured matrices, the matrix times vector product may be efficiently computed and so the main issue concerning the overall computational work in the iterative solution of a linear system with such methods is the number of iterations it takes for convergence to an acceptable accuracy. Preconditioning is usually vital to ensure that this number is kept acceptably small. Methods which guarantee some monotonic reduction in a relevant quantity at each iteration are favoured in a number of situations: the MINRES method has such a property and so is sometimes regarded as the method of choice, however the SYMMLQ method has a related 'Petrov-Galerkin' property and is favoured for reasons of numerical stability when many iterations are required (see [39]).

For a generic linear system

$$\mathcal{A}x = b$$

where $\mathcal{A}$ is symmetric (and either indefinite or definite), the MINRES method computes a sequence of iterates $\{x_k\}$ for which the residual $r_k = b - \mathcal{A}x_k$ minimizes $\|r_k\|$ over the subspace

$$r_0 + \mathrm{span}(\mathcal{A}r_0, \ldots, \mathcal{A}^k r_0). \tag{3.1}$$

The iterates themselves belong to the (shifted or affine) Krylov subspace

$$x_0 + \mathcal{K}_k(\mathcal{A}, r_0) = x_0 + \mathrm{span}(r_0, \mathcal{A}r_0, \ldots, \mathcal{A}^{k-1}r_0)$$

where x_0 is the initial iterate (the initial 'guess') and r_0 the corresponding residual. This minimization property leads immediately to a description of the convergence properties of the MINRES method: since any vector, s say, in the space (3.1) can be written as $s = q(\mathcal{A})r_0$ where q is a polynomial of degree k with constant term equal to one (ie. $q(z) = 1 + \alpha_1 z + \ldots + \alpha_k z^k$ for some coefficients α_i), we have that

$$\|r_k\| \leq \|q(\mathcal{A})r_0\| \leq \|q(\mathcal{A})\|\|r_0\|.$$

Now the diagonalization of the symmetric matrix $\mathcal{A}$ as $\mathcal{A} = X\Lambda X^T$ where Λ is the diagonal matrix of eigenvalues and the matrix X is the orthogonal matrix of eigenvectors ensures that

$$\|q(\mathcal{A})\| = \|Xq(\Lambda)X^T\| = \|q(\Lambda)\|$$

because the Euclidean norm is invariant under orthogonal transformations. Further, since $q(\Lambda)$ is a diagonal matrix we have that

$$\|r_k\| \leq \min_{q \in \Pi_k, q(0)=1} \max_{z \in \sigma(\mathcal{A})} \|q(z)\|\|r_0\|. \tag{3.2}$$

Here, Π_k is the set of (real) polynomials of degree k and $\sigma(\mathcal{A})$ is the set of eigenvalues of $\mathcal{A}$. Thus for a real symmetric matrix, convergence depends only on its eigenvalues: if there are only a few distinct eigenvalues or they are sufficiently clustered away from the origin then there are polynomials of low degree which will be small at the eigenvalues. At each additional iteration the degree increases by one and so reasonable accuracy is quickly achieved in such cases. Various constructions based on the Chebyshev polynomials can give more explicit convergence bounds, but these are somewhat less straightforward to write down for indefinite rather than definite symmetric matrices (see for example [23] or [13]).

Preconditioning correponds to the application of a matrix (or linear operator), $\mathcal{P}$ to the original linear system to yield a different linear system for which convergence of the iterative method will be significantly faster. In most situations $\mathcal{P}$ must be constructed so that it is easy/fast to solve linear systems of the form $\mathcal{P}z = r$ for z when r is given. Conceptually one can think of preconditioned iteration as applying the original iteration to

$$\mathcal{P}^{-1}\mathcal{A}\,x = \mathcal{P}^{-1}b$$

however it would in almost all cases be a really bad move to create such a nonsymmetric linear system when $\mathcal{A}$ is originally symmetric: the iterative solution of nonsymmetric linear systems is much less reliable and/or more expensive in general and most practitioners would believe that preserving symmetry is really valuable. For MINRES, a symmetric and positive definite preconditioner $\mathcal{P}$ must be employed so that we can write $\mathcal{P} = \mathcal{L}\mathcal{L}^T$ for some matrix $\mathcal{L}$ (eg. either the Cholesky factor or the matrix square root). We emphasize that this is only a mathematical artifact used to derive the method: no such factorization is required in practice. MINRES iteration can then be applied to the symmetric system

$$\mathcal{L}^{-1}\mathcal{A}\mathcal{L}^{-T}y = \mathcal{L}^{-1}b, \quad \mathcal{L}^T x = y$$

and convergence will depend on the eigenvalues of the symmetric and indefinite matrix $\mathcal{L}^{-T}\mathcal{A}\mathcal{L}^{-1}$. Via the obvious similarity tranformation

$$\mathcal{L}^{-T}\mathcal{L}^{-1}\mathcal{A}\mathcal{L}^{-T}\mathcal{L}^{T} = \mathcal{P}^{-1}\mathcal{A}$$

it is clear that the important eigenvalues are those of the matrix $\mathcal{P}^{-1}\mathcal{A}$, hence the convergence of the preconditioned MINRES iteration is described via (3.2) with the eigenvalue spectrum $\sigma(\mathcal{A})$ replaced in the preconditioned case by $\sigma(\mathcal{P}^{-1}\mathcal{A})$.

For SYMMLQ, there are similar considerations and good preconditioners should satisfy similar criteria. SQMR would generally only be used with a symmetric and indefinite preconditioner and there are no estimates of convergence in this case, though practical experience in a number of application areas indicates that SQMR convergence can be very good with a suitable indefinite preconditioner (see [18]).

In the next sections we discuss a number of possible approaches to preconditioning indefinite symmetric matices of saddle point type.

4. Block preconditioners. Block preconditioners are based more or less explicitly on the block factorization (2.1). The performance of such preconditioners depends on whether fast, approximate solvers for linear systems involving A and the Schur complement S are available [34].

Assuming that A and $-S = C + BA^{-1}B^T$ are both SPD, the ideal block diagonal preconditioner is

$$\mathcal{P}_d = \begin{bmatrix} A & O \\ O & -S \end{bmatrix}. \tag{4.1}$$

Preconditioning of $\mathcal{A}$ with $\mathcal{P}_d$ results in the matrix

$$\mathcal{M} = \mathcal{P}_d^{-1}\mathcal{A} = \begin{bmatrix} I & A^{-1}B^T \\ -S^{-1}B & O \end{bmatrix}. \tag{4.2}$$

The matrix $\mathcal{M}$ is nonsingular by assumption, is symmetrizable as described above and, as pointed out for example in [30], it satisfies

$$(\mathcal{M} - I)\left(\mathcal{M} - \frac{1}{2}(1+\sqrt{5})I\right)\left(\mathcal{M} - \frac{1}{2}(1-\sqrt{5})I\right) = O.$$

It follows that $\mathcal{M}$ is diagonalizable and has only three distinct eigenvalues, namely $1, \frac{1}{2}(1+\sqrt{5})$, and $\frac{1}{2}(1-\sqrt{5})$. Hence for each initial residual r_0, $\dim \mathcal{K}_{n+m}(\mathcal{M}, r_0) \leq 3$, which means that MINRES applied to the preconditioned system with preconditioner $\mathcal{P}_d$ will terminate after at most three steps.

Similarly, the ideal block triangular preconditioner is

$$\mathcal{P}_t = \begin{bmatrix} A & B^T \\ O & \pm S \end{bmatrix}. \tag{4.3}$$

Choosing the minus sign in (4.3) results in a diagonalizable preconditioned matrix with only two distinct eigenvalues equal to ± 1. Choosing the plus sign yields a preconditioned matrix with all the eigenvalues equal to 1; this matrix is non-diagonalizable, but has minimum polynomial of degree two. For either choice of the sign in (4.3), the non-symmetric iterative solver GMRES [36] is guaranteed to converge in at most two steps in exact arithmetic.

Obviously, the ideal preconditioners $\mathcal{P}_d$ and $\mathcal{P}_t$ are not practical, since the exact Schur complement S is generally a dense matrix and is not available. In practice, A and S are replaced by some approximations, $\hat{A} \approx A$ and $\hat{S} \approx S$. If these approximations are chosen appropriately, the preconditioned matrices have most of their eigenvalues clustered around the eigenvalues of the ideally preconditioned matrices $\mathcal{P}_d^{-1}\mathcal{A}$ and $\mathcal{P}_t^{-1}\mathcal{A}$. Clearly, the choice of the approximations $\hat{A}$ and $\hat{S}$ is highly problem-dependent. Frequently $\hat{A}$ and $\hat{S}$ are not explicitly available matrices; rather, a prescription for computing the action of $\hat{A}^{-1}$ and $\hat{S}^{-1}$ on given vectors is given. For example, in mixed finite element formulations for incompressible flow problems the block A represents a discretization of a second-order elliptic operator, and the action of $\hat{A}^{-1}$ on a vector can be computed by performing one or more iterations of some multigrid scheme. The construction of good approximations $\hat{S}$ to the Schur complement S is generally less straightforward and is highly problem-dependent; see [37, 13] for a detailed treatment in the case of incompressible flow problems.

Application of these techniques to more general saddle point problems arising in constrained optimization is more problematic. In particular, in the absence of well-understood elliptic operators it is unclear how to construct suitable approximations $\hat{A} \approx A$ and $\hat{S} \approx S$. One possibility is to use incomplete factorizations of A to build $\hat{A}$, but it is unclear how to construct good approximations to the (typically dense) Schur complement S. Section 6 describes an alternative approach that has been applied succesfully in optimization.

We conclude this section with a brief discussion of a possible connection between block preconditioners based on approximate Schur complements and model order reduction of time-invariant linear dynamical systems. Following [17, Section 4.3], the (symmetric) transfer function of certain RCL subcircuits is the $m \times m$ matrix-valued rational function

$$H(s) = B\,(sE - A)^{-1}B\,, \quad \text{where} \quad A = A^T\,, \quad E = E^T \quad \text{and} \quad s \in \mathbb{C}\,. \tag{4.4}$$

In practice, n can be in the millions while m is of the order of a few hundreds or smaller. The goal of model order reduction is to find $m \times m$ approximations to the transfer function (4.4) of the form

$$\hat{H}(s) = \hat{B}\,(s\hat{E} - \hat{A})^{-1}\hat{B}, \quad \text{where} \quad \hat{A} = \hat{A}^T, \quad \hat{E} = \hat{E}^T\,, \tag{4.5}$$

where the order $\hat{n}$ of $\hat{A}$ and $\hat{E}$ is now small, typically of the same order as m. Furthermore, the approximate transfer function $\hat{H}(s)$ must preserve certain properties of the original function $H(s)$ for the reduced-order model to be useful. A number of techniques have been developed to efficiently construct such approximations, including matrix Padé and Padé-type approximants. The approximants can be computed by means of (block) Lanczos methods; we refer the reader to [17] for a survey. These techniques have proved very effective in practice, and it would be interesting to investigate their use in constructing approximate Schur complements $\hat{S} = \hat{B}\hat{A}^{-1}\hat{B}^T \approx BA^{-1}B^T$. The approximate Schur complement could be used in turn to construct a block diagonal or block triangular preconditioner.

5. Augmented Lagrangian formulations. The assumption that A is nonsingular may be too restrictive, and indeed A is singular in many applications. However, it is often possible to use augmented Lagrangian techniques [6, 15, 16] to replace the original saddle point system with an equivalent one having the same solution but in which the $(1,1)$ block A is now nonsingular. Thus, block diagonal and block triangular preconditioners based on appproximate Schur complement techniques may still be applicable. The augmented Lagrangian idea can also be useful in cases where the $(1,1)$ block is highly ill-conditioned and in order to trasform the original saddle point system into one that is easier to precondition.

The idea is to replace the original saddle point system (1.3) with the equivalent one

$$\begin{bmatrix} A + B^TWB & B^T \\ B & O \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f + B^TWg \\ g \end{bmatrix}. \tag{5.1}$$

The $m \times m$ matrix W, to be suitably determined, is symmetric positive semidefinite. The simplest choice is to take $W = \gamma I_m$ ($\gamma > 0$). In this case the $(1,1)$ block in (5.1) is nonsingular, and indeed positive definite, provided that A is positive definite on the null space of B. The goal is to choose W so that system (5.1) is easier to solve than the original one, particularly when using iterative methods. The choice of W is highly problem-dependent; see, e.g., [4, 20] for discussions of this issue in different settings.

It is important to keep in mind that there may be a trade-off between properties of the $(1,1)$ block and properties of the augmented system (5.1). Consider for instance the case where $W = \gamma I_m$. Then a possible preconditioner for (5.1) is given by

$$\mathcal{P}_\gamma = \begin{bmatrix} A + \gamma B^T B & B^T \\ O & -\gamma^{-1} I_m \end{bmatrix}.$$

It can be shown that the quality of this preconditioner increases as γ tends to infinity; for large values of γ, however, the $(1,1)$ block becomes increasingly ill-conditioned. This is clear when one observes that for large γ the dominating term in the $(1,1)$ block becomes $\gamma B^T B$, a singular matrix with a null space of dimension $n - m$. In practice, linear systems involving $A + \gamma B^T B$ will be solved inexactly, typically by some inner iteration, and finding efficient approximate solvers may become very difficult for large values of γ. It is therefore important to strike a balance between the rate of convergence of the outer (preconditioned) iteration and the need for efficient approximate solution of linear systems involving $A + \gamma B^T B$.

Augmented Lagrangian techniques have been in use for many years in constrained optimization problems. Recent work indicates that the augmented Lagrangian approach may lead to powerful preconditioners for challenging problems in computational fluid mechanics and computational electromagnetics; see in particular [4] and [24].

6. Constraint preconditioning. The second main type of preconditioner for saddle point problems are of the general form

$$\mathcal{P} = \begin{bmatrix} H & B^T \\ B & O \end{bmatrix} \tag{6.1}$$

where $H \in \mathbb{R}^{n \times n}$ ([29, 27]). Since such an indefinite preconditioning matrix is itself a saddle point matrix which corresponds to a different quadratic energy but the same constraints as the original problem, it is called a 'constraint preconditioner'.

It is not evident that it is any easier to solve systems with this form of preconditioner than with the original matrix $\mathcal{A}$ in (1.3); since one such solution is required at each iteration this is a real issue. We will come back to this below, but firstly indicate what is known about the effect on iterative convergence of the use of preconditioners of the form (6.1).

The first point to notice is that the use of an indefinite preconditioner precludes the simple use of MINRES which requires a definite preconditioner. However a key observation is that by using the same constraint blocks in the preconditioner, the Hestenes–Stiefel Conjugate Gradient algorithm can be used: this is because solution of (1.3) with a preconditioner of the form (6.1) is equivalent to the solution of the positive definite symmetric system which would be derived by explicit elimination of the constraints with a positive definite symmetric preconditioner derived by direct elimination of these same constraints ([21]). This is a very attractive property since the Conjugate Gradient method is well known to be a very effective method with appropriate preconditioning for symmetric and positive definite systems. We emphasize that a constraint preconditioner *is required* here—for example it is clear that if no preconditioning were employed then Conjugate Gradients would not be a robust method for the indefinite saddle point system. Another consequence is that iterates for the primal variable u only are computed, so that the stopping criteria must reflect this. The Lagrange multipliers can be recovered if desired.

Thus the use of a constraint preconditioner with CG ensures (in exact arithmetic) that all of the iterates satisfy the constraints—only by employing a constraint preconditioner is this guarenteed. This appears to be a very desirable property in the context of Optimization when linear system solves are usually an inner part of an outer iterative optimization algorithm.

Given the equivalence to a symmetric positive definite problem, one might anticipate some special structure in the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$; what is perhaps not expected is that this matrix should generically be non-diagonalizable! As shown in [27] this is always the case, but this is only due to a high multiplicity eigenvalue at 1: this eigenvalue has algebraic multiplicity $2m$ but only m independent eigenvectors. In the language of canonical forms, the Jordan form of this matrix has m 2×2 diagonal blocks. This means that $\mathcal{P}^{-1}\mathcal{A}-I$ has only an $m-$dimensional kernel, but $(\mathcal{P}^{-1}\mathcal{A}-I)^2$ has the full $2m-$dimensional kernel corresponding the the eigenvalue at 1. This is highly attractive from the standpoint of Krylov subspace iteration since only two iterations will eliminate the error in a $2m-$dimensional subspace.

The outcome is that iterative convergence depends on how well H approximates A in an $n-m$-dimensional subspace with only an additional two iterations required for the eigenvalue at 1.

Returning to the solution of systems with a constraint preconditioner, there are special situations where specific orthogonality properties enable easy solution: see for example [33]. A general approach, however, involves *not* preselecting the block H, but rather choosing it in an implicit fashion. One key approach is that based on Schilders' Factorization (see[12, 9]); the idea is as follows. The factorization

$$\mathcal{P} = \begin{bmatrix} {B_1}^T & O & L_1 \\ {B_2}^T & L_2 & E \\ O & O & I \end{bmatrix} \begin{bmatrix} D_1 & O & I \\ O & D_2 & O \\ I & O & O \end{bmatrix} \begin{bmatrix} B_1 & B_2 & O \\ O & {L_2}^T & O \\ {L_1}^T & E^T & I \end{bmatrix}, \tag{6.2}$$

is exact for

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & O \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & B_1^T \\ A_{2,1} & A_{2,2} & B_2^T \\ B_1 & B_2 & O \end{bmatrix}$$

with $A_{1,1}, B_1 \in \mathbb{R}^{m\times m}$ (and other blocks correspondingly) when

$$\begin{aligned} D_1 &= {B_1}^{-T}A_{1,1}{B_1}^{-1} - {L_1}^T{B_1}^{-1} - {B_1}^{-T}L_1, \\ D_2 &= {L_2}^{-1}(A_{2,2} - {B_2}^T D_1 B_2 - EB_2 - {B_2}^T E^T){L_2}^{-T}, \\ E &= A_{2,1}{B_1}^{-1} - {B_2}^T D_1 - {B_2}^T {L_1}^T {B_1}^{-1}, \end{aligned}$$

but more importantly in our context, *any* choice of D_1, L_1 and E and any nonsingular choice of D_2, L_2 gives rise to a matrix of the form (6.1), i.e., gives rise to a constraint preconditioner in a reordered block triangular factored form. In this way by making choices for the blocks D_i, L_i and E in the factors in (6.2) a constraint preconditioner with an *implicitly defined* $(1,1)$ block H is obtained in a form in which solutions to preconditioner systems can easily be computed. The simplest choice would be

$$\begin{bmatrix} O & O & B_1^T \\ O & I & B_2^T \\ B_1 & B_2 & O \end{bmatrix} = \begin{bmatrix} {B_1}^T & O & O \\ {B_2}^T & I & O \\ O & O & I \end{bmatrix} \begin{bmatrix} O & O & I \\ O & I & O \\ I & O & O \end{bmatrix} \begin{bmatrix} B_1 & B_2 & O \\ O & I & O \\ O & O & I \end{bmatrix}. \tag{6.3}$$

It can be seen that it is always necessary to be able to compute the action of B_1^{-1}, thus it is important to be able to find a non-singular $m\times m$ leading block of the constraint matrix $B \in \mathbb{R}^{m\times n}$ possibly by reordering. A direct method (even for a dense system) will require $\mathcal{O}(m^3)$ computer (floating point) operations to achieve this, but sparsity will reduce this estimate considerably—and then the exact choice of which columns of B to reorder into B_1 also is likely to have an effect. There have been particular choices suggested for the special but important case of saddle point systems arising from interior point Optimization algorithms where large penalty parameters arise at least as convergence is approached (see [8]).

We comment that constraint preconditioners and Schilders-like factorisations for regularized saddle point systems of the form

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \tag{6.4}$$

where C is symmetric and positive semi-definite have also been described (see [10, 11]).

7. Other techniques. Most preconditioning techniques that have been proposed in the literature on saddle point problems can be reduced to one of the main classes of methods described in the three sections above. For instance, the classical Uzawa method can be shown to be a special type of block triangular preconditioner. Similarly, preconditioning methods based on null-space (or *dual variable*) formulations, see for example [1], are closely related to constraint preconditioning. An exception is represented by the *HSS preconditioner* described in [2] and further analyzed in [38]. This preconditioner is based on the nonsymmetric formulation

$$\begin{bmatrix} A & B^T \\ -B & C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix}, \quad \text{or} \quad \widehat{\mathcal{A}} x = \hat{b}. \tag{7.1}$$

Here we assume that A and C are symmetric positive semidefinite. We have the following splitting of $\hat{\mathcal{A}}$ into its symmetric and skew-symmetric parts:

$$\widehat{\mathcal{A}} = \begin{bmatrix} A & B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} A & O \\ O & C \end{bmatrix} + \begin{bmatrix} O & B^T \\ -B & O \end{bmatrix} = \mathcal{H} + \mathcal{K}. \tag{7.2}$$

Note that $\mathcal{H}$, the symmetric part of $\widehat{\mathcal{A}}$, is symmetric positive semidefinite since both A and C are. Let $\alpha > 0$ be a parameter. Similar in spirit to the classical ADI (Alternating-Direction Implicit) method, we consider the following two splittings of $\widehat{\mathcal{A}}$:

$$\widehat{\mathcal{A}} = (\mathcal{H} + \alpha\mathcal{I}) - (\alpha\mathcal{I} - \mathcal{K}) \qquad \text{and} \qquad \widehat{\mathcal{A}} = (\mathcal{K} + \alpha\mathcal{I}) - (\alpha\mathcal{I} - \mathcal{H}).$$

Here $\mathcal{I}$ denotes the identity matrix of order $n + m$. The stationary HSS iteration is then

$$x_{k+1} = x_k + \mathcal{P}_\alpha^{-1} r_k, \quad r_k = \hat{b} - \widehat{\mathcal{A}} x_k\,,$$

where the matrix $\mathcal{P}$ is given by

$$\mathcal{P} \equiv \mathcal{P}_\alpha = \tfrac{1}{2\alpha}(\mathcal{H} + \alpha\mathcal{I})(\mathcal{K} + \alpha\mathcal{I}). \tag{7.3}$$

Assuming that A is SPD and B has full rank, it has been shown in [2] that the iterative process (7.3) is convergent to the unique solution of (7.1) for all $\alpha > 0$. However, the rate of convergence of the HSS iteration is rather slow, even with the "optimal" choice of α. For these reasons it was proposed in [2] that GMRES or other Krylov subspace methods should be used to accelerate the convergence of the HSS method. In other words, the HSS method is best used as a preconditioner for (say) GMRES rather than as a stationary iterative method. Note that as a preconditioner we can use $\mathcal{P}_\alpha = (\mathcal{H} + \alpha\mathcal{I})(\mathcal{K} + \alpha\mathcal{I})$ instead of the expression given in (7.3), since the factor $\frac{1}{2\alpha}$ has no effect on the preconditioned system. The spectral analysis of HSS preconditioning for general saddle point problems can be found in [38] and [5]. The analysis shows that the eigenvalues of the preconditioned matrix are all real and positive for all $\alpha > 0$, and furthermore as $\alpha \to 0$ they all fall within two small intervals $(0, \varepsilon_1)$ and $(2 - \varepsilon_2, 2)$, with $\varepsilon_1, \varepsilon_2 > 0$ and $\varepsilon_1, \varepsilon_2 \to 0$ as $\alpha \to 0$. This suggests that α should be taken to be small, but not too small; experience suggests that for a problem scaled so that A and C have unit nonzero diagonal entries, a value of α between 0.1 and 0.5 is often a good choice. In practice, solves with the shifted matrices $\mathcal{H}+\alpha\mathcal{I}$ and $\mathcal{K} + \alpha\mathcal{I}$ are performed inexactly for efficiency reasons. Approximately solving linear systems involving $\mathcal{H} + \alpha\mathcal{I}$ is usually straightforward, whereas solving linear systems involving the shifted skew-symmetric part $\mathcal{K} + \alpha\mathcal{I}$ is slighly more complicated. This step requires the solution of a linear system of the form

$$\begin{cases} \alpha\, u_{k+1} + B^T p_{k+1} = f_k\,, \\ -B\, u_{k+1} + \alpha\, p_{k+1} = g_k\,. \end{cases} \tag{7.4}$$

This can be accomplished by first eliminating u_{k+1} from the second equation using the first one (Schur complement reduction), leading to a smaller (order m) linear system of the form

$$(BB^T + \alpha^2 I)\, p_{k+1} = B\, f_k + \alpha\, g_k\,. \tag{7.5}$$

This is a linear system with an SPD coefficient matrix which can be approximately solved by, e.g., a preconditioned Conjugate Gradient method. In this case, it is necessary to use a flexible Krylov subspace method, such as FGMRES, for the outer iteration; see [35].

8. Numerical examples. We firstly present an example of block diagonal preconditioning for a problem in incompressible fluid mechanics.

The underlying problem is the Stokes problem which is the particular case $\sigma = 0$ of the *generalized Stokes problem*:

$$\sigma\mathbf{u} - \nu\nabla^2\mathbf{u} + \nabla\, p = \mathbf{f} \quad \text{in } \Omega \tag{8.1}$$

$$\operatorname{div}\mathbf{u} = 0 \quad \text{in } \Omega \tag{8.2}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega\,. \tag{8.3}$$

Here $\mathbf{u}$ is the velocity and p the pressure (the Lagrange multiplier in this application). $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ is the domain of the partial differential equation with boundary $\partial\Omega$ on which we have assumed simple Dirichlet conditions. The parameter ν is the kinematic viscosity which is taken to have the value one for the classical Stokes problem. See [13] for details.

This first example is computed with a common mixed finite element formulation: the block preconditioner combines a single simple multigrid V-cycle approximation of A and a diagonal matrix to approximate S and is run using the freely available IFISS software ([26]). We include iteration counts (which are seen to be essentially constant—indeed to reduce slightly—over a range of increasing problem dimension) and cpu times on the same workstation. Timings for a direct solution are given for comparison.

Table 8.1

Block dimensions and number of MINRES iterations needed for 10^{-6} reduction in residual for locally stabilized $Q1 - P0$ mixed finite elements for Stokes flow in a cavity. Block diagonal preconditioner: $\hat{A}$ is one multigrid V-cycle with 1,1 relaxed Jacobi smoothing and $\hat{S}$ is the diagonal pressure mass matrix. The cpu time (in seconds) is that required on the same computer (a Sun sparcv9 502 MHz processor with 1024 Mb of memory). The cpu time is also given for a sparse direct solve (UMFPACK in MATLAB).

grid	n	m	iterations	cpu time	sparse direct cpu
64×64	8450	4096	38	14.3	6.8
128×128	33282	16384	37	37.7	48.0
256×256	132098	65536	36	194.6	897
512×512	526339	263169	35	6903	out of memory

We can notice from Table 8.1 that for the largest-dimensional problem memory becomes an issue: the sparse direct method runs out of memory completely and fails for this problem and the timing for the iterative method is much greater than expected presumably because of slower memory access times for the more remote levels of cache which are needed for this problem.

To give an example of constraint preconditioning, we turn to problems from Optimization, specifically to a family of test problems from the CUTEr test set ([22]). We present results only for the simplest Schilders' factorization (6.3) for three of the family of CVXQP1 test problems. As indicated in the section above, Conjugate Gradient iteration is applicable with constraint preconditioning and this is applied here. The number of Conjugate Gradient iterations to achieve a 10^{-6} reduction in the preconditioned residual (defined only on the $n-$dimensional space of the primal variable u as described above) are given in Table 8.2.

Table 8.2

Block dimensions and number of Conjugate Gradient iterations needed for 10^{-6} reduction in the preconditioned residual for the simplest Schilders' factorization preconditioner (6.3).

test problem	n	m	iterations
CVXQP1_S	100	50	44
CVXQP1_M	1000	500	28
CVXQP1_L	10000	5000	10

Our final numerical example demonstrates the performance of the HSS preconditioner on the generalized Stokes problem.

In Table 8.3 we report the numerical results for Flexible GMRES with inexact HSS preconditioning applied to a set of generalized Stokes problems. The discrete saddle point problems were generated in this case by the Marker-and-Cell (MAC) finite difference discretization on a $40 \times 40 \times 40$ grid for different values of σ ($= 1/\Delta t$ in the context of implicit solution of time-dependent problems) and ν. Homogeneous Dirichlet boundary conditions were imposed on the velocities. Here $\Omega = [0, 1] \times [0, 1] \times [0, 1]$; the discrete problem has over 250,000 unknowns. The parameter α was set to 0.5, and a zero initial guess was used. The outer iteration was stopped when a reduction of the initial residual by six orders of magnitude was reached. For the inexact inner solves we used Conjugate Gradients with incomplete Cholesky preconditioning; the inner iterations were stopped as soon as a reduction of the initial residual by one order of magnitude was attained. This only required 1-2 PCG iterations per inner linear solve. The iteration counts, which can be shown to be largely independent of the grid size, improve for increasing σ and decreasing ν.

Table 8.3
Iteration count for 3D generalized Stokes problem, inexact HSS preconditioning.

σ	$\nu = 0.1$	$\nu = 0.01$	$\nu = 0.001$	$\nu = 10^{-6}$
1	45	27	16	13
10	32	19	15	12
20	30	18	14	11
50	28	15	13	11
100	25	14	12	10

In Table 8.4 we show timings (in seconds) for an unsteady Stokes problem with $\nu = 0.001$ for different grids. Denoted by h the grid size, we let $\sigma = h^{-1}$. We use HSS preconditioning with $\alpha = 0.5$. We also report the dimensions n and m and the total number of FGMRES iterations. The test runs were done on one processor of a SunFire V880 workstation with 8 CPUs and 16 GB of memory.

Table 8.4
Results for 3D unsteady Stokes problem, $\nu = 0.001$.

grid	n	m	iterations	cpu time
$10 \times 10 \times 10$	2700	1000	12	0.42
$20 \times 20 \times 20$	22800	8000	12	4.66
$30 \times 30 \times 30$	78300	27000	12	20.97
$40 \times 40 \times 40$	187200	64000	13	66.02

9. Conclusions. Saddle point problems arise naturally in many large scale computations, particularly in the solution of PDEs by mixed finite elements, interior point methods for constrained optimization, weighted least squares, and so forth. The last decade has seen considerable progress in the development of iterative solvers and preconditioners for this class of problems. In this paper we have given a concise overview of some of the most promising preconditioning techniques for linear systems in saddle point form, in particular block and constraint preconditioning. We have also pointed out a possible connection between preconditioners based on approximate Schur complements and the approximation of matrix-valued transfer functions, an essential component of model order reduction for time-invariant linear dynamical systems.

Acknowledgements. This paper originated in the stimulating atmosphere fostered by Wil Schilders and Henk van der Vorst at the Workshop on Model Reduction, Coupled Problems and Optimization held in Leiden, The Netherlands, on September 19–23, 2005. We would like to thank Wil and Henk for inviting us to the Workshop and for suggesting that we write the present overview.

REFERENCES

[1] M. Arioli, J. Mariška, M. Rozložník, and M. Tůma, *Dual variable methods for mixed-hybrid finite element approximation of the potential fluid flow problem in porous media*, Electr. Trans. Numer. Anal., 22 (2006), pp. 17–40.

[2] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.

[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[4] M. Benzi and M. A. Olshanskii, *An augmented Lagrangian approach to the Oseen problem*, Technical Report TR-2005-013-A, Department of Mathematics and Computer Science, Emory University, 2005.

[5] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., to appear.

[6] P. Bochev and R. B. Lehoucq, *Regularization and stabilization of discrete saddle-point variational problems*, Electr. Trans. Numer. Anal., 22 (2006), pp. 97–113.

[7] D. Braess, *Finite Elements. Theory, Fast Solvers, and Applications in Solid Mechanics.* Second Edition, Cambridge University Press, 2001.

[8] H. S. Dollar, *Iterative Linear Algebra for Constrained Optimization*, DPhil (PhD) thesis, Oxford University Computing Laboratory, 2005.

[9] H. S. Dollar, N. I. M. Gould, and A. J. Wathen, *On implicit-factorization constraint preconditioners*, to appear in procedings of 'Large Scale Nonlinear Optimization', Erice, Italy, June 2004, Springer.

[10] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., to appear.

[11] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Using constraint preconditioners with regularized saddle-point problems*, Comput. Optim. Appl., submitted, 2005.

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

[13] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2005.

[14] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner, Chichester and Stuttgart, 1996.

[15] R. Fletcher, *Practical Methods of Optimization (Second Edition)*, J. Wiley & Sons, Chichester, 1987.

[16] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Application to the Solution of Boundary-Value Problems*, Stud. Math. Appl., Vol. 15, North-Holland, Amsterdam, 1983.

[17] R. W. Freund, *Model reduction based on Krylov subspaces*, Acta Numerica, 12 (2003), pp. 267–319.

[18] R. W. Freund and N. M. Nachtigal, *A new Krylov-subspace method for symmetric indefinite linear systems*, in proceedings of 14th IMACS World Congress on Computational and Applied Mathematics, W. F. Ames, ed., IMACS, 1994, pp. 1253–1256.

[19] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1995), pp. 319–341.

[20] G. H. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.

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

[22] N. I. M. Gould, D. Orban, and Ph. L. Toint, *CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, Revisited*, Tech. Report TR/PA/01/04, CERFACS, Toulouse, France, 2001.

[23] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[24] C. Greif and D. Schötzau, *Preconditioners for the discretized time-harmonic Maxwell equations in mixed form*, submitted.

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

[26] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: a Matlab toolbox for modelling incompressible flow*, Manchester University Numerical Analysis Report No. 474 (2005), submitted to ACM Transactions on Mathematical Software.

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

[28] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential*

and integral operators, J. Res. Nat. Bur. Stand., 45 (1950), pp. 255-282.
[29] L. LUKŠAN AND J. VLČEK, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.
[30] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
[31] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[32] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 1999.
[33] I. PERUGIA AND V. SIMONCINI, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.
[34] T. RUSTEN AND R. WINTHER, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.
[35] Y. SAAD, *Iterative Methods for Sparse Linear Systems, Second Edition*, SIAM, Philadelphia, PA, 2003.
[36] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[37] D. J. SILVESTER AND A. J. WATHEN, *Fast iterative solution of stabilised Stokes systems. II: Using general block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.
[38] V. SIMONCINI AND M. BENZI, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 377–389.
[39] G. L. G. SLEIJPEN, H. A. VAN DER VORST, AND J. MODERSITZKI, *Effects of rounding errors in determining approximate solutions in Krylov solvers for symmetric indefinite linear systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 726-751.
[40] G. STRANG, *Introduction to Applied Mathematics*, Wellesley, Cambridge, MA, 1986.
[41] H. A. VAN DER VORST, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK, 2003.