BLOCK-DIAGONAL AND CONSTRAINT PRECONDITIONERS FOR NONSYMMETRIC INDEFINITE LINEAR SYSTEMS. PART I: THEORY[*]

ERIC DE STURLER[†] AND JÖRG LIESEN[‡]

Abstract. We study block-diagonal preconditioners and an efficient variant of constraint preconditioners for general two-by-two block linear systems with zero (2,2)-block. We derive block-diagonal preconditioners from a splitting of the (1,1)-block of the matrix. From the resulting preconditioned system we derive a smaller, so-called *related system* that yields the solution of the original problem. Solving the related system corresponds to an efficient implementation of constraint preconditioning. We analyze the properties of both classes of preconditioned matrices, in particular their spectra. Using analytical results, we show that the related system matrix has the more favorable spectrum, which in many applications translates into faster convergence for Krylov subspace methods. We show that fast convergence depends mainly on the quality of the splitting, a topic for which a substantial body of theory exists. Our analysis also provides a number of new relations between block-diagonal preconditioners and constraint preconditioners. For constrained problems, solving the related system produces iterates that satisfy the constraints exactly, just as for systems with a constraint preconditioner. Finally, for the Lagrange multiplier formulation of a constrained optimization problem we show how scaling nonlinear constraints can dramatically improve the convergence for linear systems in a Newton iteration. Our theoretical results are confirmed by numerical experiments on a constrained optimization problem.

We consider the general, nonsymmetric, nonsingular case. Our only additional requirement is the nonsingularity of the Schur-complement–type matrix derived from the splitting that defines the preconditioners. In particular, the (1,2)-block need not equal the transposed (2,1)-block, and the (1,1)-block might be indefinite or even singular. This is the first paper in a two-part sequence. In the second paper we will study the use of our preconditioners in a variety of applications.

Key words. saddle point systems, indefinite systems, eigenvalue bounds, Krylov subspace methods, preconditioning, constrained optimization, mesh-flattening

AMS subject classifications. 65F10, 65F15, 65D18

DOI. 10.1137/S1064827502411006

1. Introduction. We study preconditioners for general nonsingular linear systems of the type

$$\mathcal{A}u = \left[\begin{array}{cc} A & B^T \\ C & 0 \end{array}\right] \left[\begin{array}{c} \tilde{x} \\ \tilde{y} \end{array}\right] = \left[\begin{array}{c} \tilde{f} \\ \tilde{g} \end{array}\right],$$

(1.1) $\qquad A \in \mathbb{R}^{n \times n}, \quad B, C \in \mathbb{R}^{m \times n}, \ \text{with} \ n \geq m.$

Such systems arise in a large number of applications, for example, the (linearized) Navier–Stokes equations and other physical problems with conservation laws as well

as constrained optimization problems.

As such systems are typically large and sparse, solution by iterative methods has been studied extensively. Much attention has focused on the Navier–Stokes problem; see, e.g., [8, 16, 31, 32, 35]. The techniques for solving systems like (1.1) are so numerous that it is almost impossible to give an overview. In addition to the methods developed specifically for Navier–Stokes problems, existing techniques also include Uzawa-type algorithms [9], splitting schemes [6, 12], constraint preconditioning [12, 17, 21, 24, 25, 26], and (preconditioned) Krylov subspace methods based on (approximations to) the Schur complement [1, 18, 23].

We start with block-diagonal preconditioners for the general system (1.1); see section 2 for our assumptions. Results for the general system have been obtained before; for example, Murphy, Golub, and Wathen [23] propose the block-diagonal preconditioner

$$\begin{bmatrix} A^{-1} & 0 \\ 0 & (CA^{-1}B^T)^{-1} \end{bmatrix}.$$

If defined, this preconditioner leads to (left or right) preconditioned matrices that are diagonalizable and have at most three distinct eigenvalues. Hence, a Krylov subspace method with the optimality or the Galerkin property—e.g., GMRES [27] or BiCG [10]—will converge in at most three steps [23, Remark 3]. However, this preconditioner is more expensive than direct solution by block elimination. Thus, one typically uses approximations to A^{-1} and $(CA^{-1}B^T)^{-1}$.

We derive such approximations from a splitting of the (1,1)-block, $A = D - E$, where D can be efficiently inverted. Then, from a splitting of the preconditioned matrix we derive a fixed point iteration and its so-called related system [15] that have (significantly) fewer unknowns. We provide a careful eigenvalue analysis for the preconditioned system and the related system. This analysis shows that the related system is not only smaller than the preconditioned system but typically also leads to faster convergence for the GMRES algorithm. Furthermore, as solving the related system corresponds to an efficient implementation of constraint preconditioning, each GMRES iterate for the related system satisfies the constraints, $C\tilde{x} = \tilde{g}$, in (1.1) exactly. For many applications this is important, since a solution (accurate or less accurate) is only meaningful if the constraints are satisfied. Hence, this approach avoids the need to solve the problem very accurately just to satisfy the constraints, and may lead to significant savings. For a special starting guess, a similar result is given in [24], but the potential savings of solving a much smaller system are not elaborated. Finally, for the Lagrange multiplier formulation of a constrained optimization problem we show how scaling nonlinear constraints can dramatically improve the convergence for linear systems in a Newton iteration.

This paper is the first of a two-part sequence. Here, we focus on the derivation and analysis of our preconditioned iterations and succinctly demonstrate our results numerically. In the second paper [19] we demonstrate our results on a variety of applications and discuss efficient implementations of our preconditioners.

The paper is organized as follows. In section 2 we introduce the block-diagonal preconditioners and the resulting preconditioned systems. In section 3 we study the properties of the preconditioned matrices, in particular their eigendecompositions. In section 4 we derive the fixed point iteration and its related system. We also analyze the spectral radius of the fixed point iteration matrix and the spectrum of the related system matrix. In section 5 we discuss why typically it is more efficient to apply

GMRES to the related system than to the preconditioned system. In section 6 we introduce the application for our numerical experiments, and we discuss the scaling of nonlinear constraints in this constrained optimization problem to improve the convergence in each Newton step. In section 7 we show a few numerical results, and in section 8 we give our conclusions.

2. Block-diagonal preconditioners. Suppose that a system of the form (1.1) is given, and that we split the (1,1)-block of $\mathcal{A}$ into

$$(2.1) \qquad\qquad A \; = \; D - E,$$

where D is invertible. Our *only assumptions* are that $\mathcal{A}$ and $CD^{-1}B^T$ are invertible.

Since D and $CD^{-1}B^T$ are both invertible, we can define the preconditioner

$$(2.2) \qquad\qquad \mathcal{P}(D) \; = \; \left[\begin{array}{cc} D^{-1} & 0 \\ 0 & (CD^{-1}B^T)^{-1} \end{array} \right].$$

Multiplying $\mathcal{A}$ from the left or right by $\mathcal{P}(D)$ results in the matrices

$$(2.3) \qquad \left[\begin{array}{cc} I_n - D^{-1}E & D^{-1}B^T \\ (CD^{-1}B^T)^{-1}C & 0 \end{array} \right] \; \text{or} \; \left[\begin{array}{cc} I_n - ED^{-1} & B^T(CD^{-1}B^T)^{-1} \\ CD^{-1} & 0 \end{array} \right],$$

respectively. Both of these matrices are of the form

$$(2.4) \qquad \mathcal{B}(S) \; = \; \left[\begin{array}{cc} I_n - S & N \\ M & 0 \end{array} \right], \; \text{where}$$

$$(2.5) \qquad MN = I_m, \;\; (NM)^2 = NM, \;\; S \in \mathbb{R}^{n \times n}, \;\; M, N^T \in \mathbb{R}^{m \times n}, \;\; n \geq m.$$

After applying $\mathcal{P}(D)$ to (1.1), we are interested in solving linear systems of the form

$$(2.6) \qquad\qquad \mathcal{B}(S) \left[\begin{array}{c} x \\ y \end{array} \right] \; = \; \left[\begin{array}{c} f \\ g \end{array} \right].$$

> *Note that either the vector $[\tilde{x}^T, \tilde{y}^T]^T$ or $[\tilde{f}^T, \tilde{g}^T]^T$ from the original problem (1.1) is modified to account for the application of either right or left preconditioning.*

Remark 2.1. If $n = m$, the matrix $CD^{-1}B^T$ is invertible if and only if both C and B^T are invertible. In this case we can solve (1.1) directly by computing $x = C^{-1}g$ and $y = B^{-T}(f - Ax)$. This has essentially the same cost as one multiplication of (1.1) by (2.2), and preconditioning has no advantage over solving (1.1) directly. While many of our results hold true for $n = m$, we consider this case of little interest.

Our general approach is to consider which splittings $A = D - E$ result in preconditioned systems (2.6) that are solved efficiently by an iterative method. If we consider only the iteration count, the most effective preconditioner of the form (2.2) is derived by Murphy, Golub, and Wathen [23]. In our notation it is $\mathcal{P}(A)$, corresponding to the trivial splitting $D = A$ and $E = 0$. As shown in [23], the left and right preconditioned matrices, both of the form $\mathcal{B}(0)$, are diagonalizable with at most three distinct eigenvalues in the nonsingular case. Hence, any Krylov subspace method with a Galerkin or optimality property—e.g., BiCG [10] or GMRES [27] (see [14] for an overview)—will converge in at most three steps. While this is an attractive feature, the preconditioner $\mathcal{P}(A)$ requires multiplications by A^{-1}. However, in

many applications we have $n \gg m$, and the computational effort to solve for A is not significantly less than the effort to solve for $\mathcal{A}$. In addition, a difficult subproblem arises in inverting the Schur complement $CA^{-1}B^T$. Since A^{-1} is usually not available explicitly, $CA^{-1}B^T$ cannot be formed without solving for A a significant number of times. Murphy, Golub, and Wathen are, of course, aware that using $\mathcal{P}(A)$ is typically prohibitive, and they remark that approximations to A^{-1} and $(CA^{-1}B^T)^{-1}$ should lead to clustered eigenvalues as well, where the clustering "depends on the quality of the approximations" [23, Remark 5]. To some extent this remark was the motivation for part of our work.

Summarizing, the general strategy must be to choose a splitting that leads to efficiently invertible matrices D and $CD^{-1}B^T$ and preserves the properties of the algebraically optimal preconditioner $\mathcal{P}(A)$ as much as possible. To derive guidelines for such choices we analyze how properties of the preconditioned matrices depend on the splitting.

3. Properties of the matrices $\mathcal{B}(S)$. Our first goal is to identify the conditions under which $\mathcal{B}(S)$ is singular.

THEOREM 3.1. *A matrix $\mathcal{B}(S)$ of the form* (2.4)–(2.5) *is singular if and only if* 1 *(one) is an eigenvalue of the matrix* $(I_n - NM)S$. *In particular, each matrix $\mathcal{B}(0)$ of the form* (2.4)–(2.5) *is nonsingular.*

Proof. The matrix $\mathcal{B}(S)$ is singular if and only if there exists a nonzero vector $[x^T, y^T]^T$ for which $\mathcal{B}(S)\,[x^T, y^T]^T = 0$. This is equivalent to the two equations

$$\text{(i) } (I_n - S)x + Ny = 0, \quad \text{(ii) } Mx = 0.$$

Equation (i) is equivalent to $Sx - Ny = x$. Inserting this into equation (ii), and using that $MN = I_m$, yields $y = MSx$. Hence $x = 0$ implies $y = 0$. Inserting $y = MSx$ into $Sx - Ny = x$ shows that x has to satisfy $(I_n - NM)Sx = x$. There exists a nonzero x satisfying this requirement; i.e., $\mathcal{B}(S)$ is singular if and only if $(I_n - NM)S$ has an eigenvalue 1. $\quad\square$

Remark 3.2. Under the assumption that the preconditioner $\mathcal{P}(A)$ exists, i.e., A and $CA^{-1}B^T$ are both invertible, Theorem 3.1 shows that the matrices $\mathcal{P}(A)\,\mathcal{A}$ and $\mathcal{A}\mathcal{P}(A)$ are always nonsingular. Hence the zero eigenvalue included in the discussion of Murphy, Golub, and Wathen (cf. [23, Remark 1]) never occurs.

The inverse of $\mathcal{B}(0)$ can be easily computed and is given by

$$(3.1) \qquad \mathcal{B}(0)^{-1} = \left[\begin{array}{cc} I_n - NM & N \\ M & -I_m \end{array} \right] = \mathcal{B}(0) - \left[\begin{array}{cc} NM & 0 \\ 0 & I_m \end{array} \right].$$

In case of the trivial splitting ($S = 0$), one can therefore simply solve (2.6) via (3.1), which yields

$$\text{(i) } x = (I_n - NM)f + Ng, \quad \text{(ii) } y = Mf - g.$$

The solution is computed using two matrix-vector products with N, one matrix-vector product with M, and three vector additions. Therefore, the cost of this solution method is comparable to that of just one step of a Krylov subspace method applied to (2.6).

Next, we study the properties of eigenvalues and eigenvectors of $\mathcal{B}(S)$. In the case $S = 0$, we can directly relate the eigenvalues and eigenvectors of $\mathcal{B}(0)$ to the projection matrix NM.

THEOREM 3.3. *Let $\mathcal{B}(0)$ be of the form* (2.4)–(2.5). *Then $\mathcal{B}(0)$ is diagonalizable, and it has the following eigenvalues and eigenvectors:*

- $n - m$ eigenpairs of the form $(1, [u_j^T, 0]^T)$, where $u_1, \ldots, u_{n-m}$ form a basis of $\mathrm{Null}(NM)$, the nullspace of NM.
- $2m$ eigenpairs of the form $(\lambda^\pm, [u_j^T, (\lambda^\pm)^{-1}(Mu_j)^T]^T)$, where $\lambda^\pm \equiv (1 \pm \sqrt{5})/2$, and $u_{n-m+1}, \ldots, u_n$ form a basis of $\mathrm{Range}(NM)$, the range of NM.

In particular, if we denote

$$(3.2) \quad U_1 \equiv [u_1, \ldots, u_{n-m}] \in \mathbb{R}^{n \times (n-m)}, \qquad U_2 \equiv [u_{n-m+1}, \ldots, u_n] \in \mathbb{R}^{n \times m},$$

then the eigenvector matrix $\mathcal{Y}(0)$ of $\mathcal{B}(0)$ is given by

$$(3.3) \qquad \mathcal{Y}(0) = \begin{bmatrix} U_1 & U_2 & U_2 \\ 0 & (\lambda^+)^{-1}MU_2 & (\lambda^-)^{-1}MU_2 \end{bmatrix},$$

where both $[U_1, U_2] \in \mathbb{R}^{n \times n}$ and $(\lambda^-)^{-1}MU_2 \in \mathbb{R}^{m \times m}$ are nonsingular.

Proof. To compute the eigendecomposition of $\mathcal{B}(0)$, we consider the equation $\mathcal{B}(0)[u^T, v^T]^T = \lambda[u^T, v^T]^T$, which is equivalent to the two equations

$$\text{(i) } u + Nv = \lambda u, \quad \text{(ii) } Mu = \lambda v.$$

Since $\mathcal{B}(0)$ is nonsingular we can assume that $\lambda \neq 0$, so that equation (ii) is equivalent to $v = \lambda^{-1}Mu$. Inserting this into (i) and multiplying the resulting equation by λ yields

$$NMu = (\lambda^2 - \lambda)u;$$

i.e., the u-component of each eigenvector $[u^T, v^T]^T$ of $\mathcal{B}(0)$ is an eigenvector of the projection NM. Hence $\lambda^2 - \lambda$ is either equal to one (i.e., $\lambda = (1 \pm \sqrt{5})/2$) or equal to zero (i.e., $\lambda = 1$).

Next note that since $MN = I_m$, we have

$$m = \mathrm{Rank}(MN) \leq \min(\mathrm{Rank}(M), \mathrm{Rank}(N)) \leq m.$$

Hence $\mathrm{Rank}(N) = m$, so that $\mathrm{Null}(NM)$ is equal to $\mathrm{Null}(M)$ and has dimension $n - m$. If $u_1, \ldots, u_{n-m}$ form a basis of $\mathrm{Null}(NM)$, then the $n - m$ pairs $(1, [u_j^T, 0]^T)$, $j = 1, \ldots, n - m$, satisfy equations (i) and (ii). Furthermore, let the m vectors $u_{n-m+1}, \ldots, u_n$ form a basis of $\mathrm{Range}(NM)$. Using these vectors in (i) and (ii) shows that the remaining $2m$ eigenpairs are $(\lambda^\pm, [u_j^T, (\lambda^\pm)^{-1}(Mu_j)^T]^T)$, $j = n - m + 1, \ldots, n$, with $\lambda^\pm \equiv (1 \pm \sqrt{5})/2$.

Finally, $[U_1, U_2]$ is nonsingular since this matrix is the eigenvector matrix of the projection NM. Furthermore, if MU_2 were singular, then a nonzero vector w would exist such that $MU_2w = 0$. However, multiplication with N yields $NMU_2w = U_2w = 0$, which is a contradiction since the columns of U_2 are linearly independent. $\quad\square$

Remark 3.4. The statement of Theorem 3.3 contains the complete eigendecompositions of the preconditioned matrices $\mathcal{P}(A)\mathcal{A}$ and $\mathcal{A}\mathcal{P}(A)$ that are the subject of [23]. In that paper Murphy, Golub, and Wathen showed that the two matrices are diagonalizable and derive the location of their eigenvalues.

Note that in the case $n = m$, we have $\mathrm{Null}(NM) = \{0\}$, and Theorem 3.3 shows that in this case the only distinct eigenvalues of $\mathcal{B}(0)$ are $(1 \pm \sqrt{5})/2$. As discussed in Remark 2.1, this case is of little interest for our purposes. *In the following we will therefore assume that $n > m$.*

Next, we derive bounds on the eigenvalues of each matrix $\mathcal{B}(S)$ in terms of the corresponding matrix $\mathcal{B}(0)$.

THEOREM 3.5. *Consider matrices $\mathcal{B}(S)$ of the form* (2.4)–(2.5) *with fixed N and M. Let $\mathcal{B}(0) = \mathcal{Y}(0)\,\mathcal{D}\,\mathcal{Y}(0)^{-1}$ denote the eigendecomposition of $\mathcal{B}(0)$, with eigenvector matrix $\mathcal{Y}(0)$ given as in* (3.3), *and let $[U_1, U_2]$ denote the corresponding eigenvector matrix of the projection NM. Then for each matrix S, and each eigenvalue λ_S of $\mathcal{B}(S)$, there is an eigenvalue λ of $\mathcal{B}(0)$ such that*

$$(3.4) \qquad |\lambda_S - \lambda| \;\leq\; \left\| \mathcal{Y}(0)^{-1} \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \mathcal{Y}(0) \right\|$$

$$(3.5) \qquad\qquad\qquad \leq\; c_S \, \| \, [U_1, U_2]^{-1}\, S\, [U_1, U_2]\, \|,$$

where $c_S \equiv \left(2 + \frac{2}{5}(\lambda^-)^2 \right)^{\frac{1}{2}} \approx 1.4672$.

Proof. A given matrix $\mathcal{B}(S)$ can be additively split into

$$(3.6) \qquad \mathcal{B}(S) \;=\; \mathcal{B}(0) - \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}.$$

Since $\mathcal{B}(0)$ is diagonalizable and has eigenvector matrix $\mathcal{Y}(0)$, inequality (3.4) follows from a well-known result in matrix perturbation theory [33, Theorem IV.1.12].

To prove (3.5) consider the two-by-two block decomposition

$$\mathcal{Y}(0) \;=\; \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix},$$

where $Y_{11} \equiv [U_1, U_2] \in \mathbb{R}^{n \times n}$ and $Y_{22} \equiv (\lambda^-)^{-1} M U_2 \in \mathbb{R}^{m \times m}$ are both invertible; cf. (3.3). Then the inverse of $\mathcal{Y}(0)$ satisfies

$$\mathcal{Y}(0)^{-1} = \begin{bmatrix} (Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})^{-1} & -Y_{11}^{-1}Y_{12}\,(Y_{22} - Y_{21}Y_{11}^{-1}Y_{12})^{-1} \\ -Y_{22}^{-1}Y_{21}\,(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})^{-1} & (Y_{22} - Y_{21}Y_{11}^{-1}Y_{12})^{-1} \end{bmatrix}.$$

An elementary computation now shows that

$$
\begin{aligned}
(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})^{-1} &= \left([U_1, U_2] - U_2(\lambda^-)(MU_2)^{-1}\,[0,(\lambda^+)^{-1}MU_2] \right)^{-1} \\
&= \left([U_1, U_2] - [0,(\lambda^-/\lambda^+)U_2] \right)^{-1} \\
&= \begin{bmatrix} I_{n-m} & 0 \\ 0 & (\lambda^+/\sqrt{5})I_m \end{bmatrix} [U_1, U_2]^{-1} \\
&\equiv \hat{I}_n\,[U_1, U_2]^{-1}, \text{ and}
\end{aligned}
$$

$$-Y_{22}^{-1}Y_{21}\,(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})^{-1} = -[0,\,(\lambda^-/\sqrt{5})I_m]\,[U_1, U_2]^{-1} \;\equiv\; \hat{I}_m\,[U_1, U_2]^{-1}.$$

Using these relations, the square of the right-hand side of (3.4) is equal to

$$\left\| \begin{bmatrix} \hat{I}_n[U_1,U_2]^{-1}S[U_1,U_2] & \hat{I}_n[U_1,U_2]^{-1}SU_2 \\ \hat{I}_m[U_1,U_2]^{-1}S[U_1,U_2] & \hat{I}_m[U_1,U_2]^{-1}SU_2 \end{bmatrix} \right\|^2$$

$$= \max_{\|[a,b]\|=1} \left\| \begin{bmatrix} \hat{I}_n[U_1,U_2]^{-1}S\,([U_1,U_2]a + U_2 b) \\ \hat{I}_m[U_1,U_2]^{-1}S\,([U_1,U_2]a + U_2 b) \end{bmatrix} \right\|^2$$

$$= \max_{\|[a_1,a_2,b]\|=1} \left\| \begin{bmatrix} \hat{I}_n[U_1,U_2]^{-1}S\,(U_1 a_1 + U_2(a_2 + b)) \\ \hat{I}_m[U_1,U_2]^{-1}S\,(U_1 a_1 + U_2(a_2 + b)) \end{bmatrix} \right\|^2$$

$$\leq \max_{\|c\|\leq\sqrt{2}} \left\| \begin{bmatrix} \hat{I}_n[U_1,U_2]^{-1}S\,[U_1,U_2]c \\ \hat{I}_m[U_1,U_2]^{-1}S\,[U_1,U_2]c \end{bmatrix} \right\|^2$$

$$\leq 2 \left(\|\hat{I}_n[U_1, U_2]^{-1} S[U_1, U_2]\|^2 + \|\hat{I}_m[U_1, U_2]^{-1} S[U_1, U_2]\|^2 \right)$$

$$\leq 2 \left(1 + (\lambda^-/\sqrt{5})^2 \right) \|[U_1, U_2]^{-1} S[U_1, U_2]\|^2.$$

Taking square roots completes the proof. □

Each choice of the splitting $A = D - E$ leads to *fixed* matrices $\mathcal{B}(0)$ and $\mathcal{B}(S)$ for which Theorem 3.5 will hold. Hence, the theorem, which allows S to vary, is more general than our application requires. Furthermore, in bounding the right-hand side of (3.4) from above by (3.5), we have used three inequalities, which should generally be tight. Thus, we can expect the right-hand side of (3.4) to be close to (3.5).

To analyze implications of Theorem 3.5, recall that the columns of U_1 and U_2 form the bases of $\text{Null}(NM)$ and $\text{Range}(NM)$, respectively. We choose both bases to be orthonormal, i.e., $U_1^T U_1 = I_{n-m}$ and $U_2^T U_2 = I_m$. A key ingredient of our analysis is the singular value decomposition (SVD) of the matrix $U_1^T U_2$,

$$(3.7) \qquad U_1^T U_2 = \Phi \Omega \Psi^T = [\varphi_1, \ldots, \varphi_{n-m}] \, \text{diag}(\omega_1, \ldots, \omega_k) \, [\psi_1, \ldots, \psi_m]^T,$$

where $\Phi \in \mathbb{R}^{(n-m) \times (n-m)}$ and $\Psi \in \mathbb{R}^{m \times m}$ are both orthogonal matrices, $\Omega \in \mathbb{R}^{(n-m) \times m}$ with $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_k$, and $k = \min(n - m, m)$. It is well known that the singular values satisfy $\omega_j = \cos(\theta_j)$, where the θ_j are the principal angles between $\text{Null}(NM)$ and $\text{Range}(NM)$; see [11, section 12.4]. Since NM is a projection, we have $\text{Null}(NM) \cap \text{Range}(NM) = \{0\}$, and thus $\omega_j \in [0, 1)$, $j = 1, \ldots, k$.

LEMMA 3.6. *Let $U_1^T U_1 = I_{n-m}$, $U_2^T U_2 = I_m$, and let the SVD (3.7) be defined. Then $[U_1, U_2]$ has $2k$ singular values $(1 \pm \omega_j)^{1/2}$, $j = 1, 2, \ldots, k$, and an $(n - 2k)$-fold singular value 1 (one), where $k = \min(n - m, m)$. In particular, the condition number of $[U_1, U_2]$ is given by*

$$(3.8) \qquad \kappa([U_1, U_2]) = \left(\frac{1 + \omega_1}{1 - \omega_1} \right)^{1/2}.$$

Proof. First assume that $n - m \geq m$; thus $k \equiv m$ (above). Since

$$[U_1, U_2]^T [U_1, U_2] = \begin{bmatrix} \Phi & 0 \\ 0 & \Psi \end{bmatrix} \begin{bmatrix} I_{n-m} & \Omega \\ \Omega^T & I_m \end{bmatrix} \begin{bmatrix} \Phi^T & 0 \\ 0 & \Psi^T \end{bmatrix},$$

the singular values of $[U_1, U_2]$ are the square roots of the eigenvalues of

$$(3.9) \qquad I_n + \begin{bmatrix} 0_{n-m} & \Omega \\ \Omega^T & 0_m \end{bmatrix} = I_n + \begin{bmatrix} 0 & 0 & \Omega_m \\ 0 & 0_{n-2m} & 0 \\ \Omega_m & 0 & 0 \end{bmatrix},$$

where $\Omega_m = \text{diag}(\omega_1, \ldots, \omega_m)$. Following [11, pp. 448–449], we note that this matrix has as its eigenvalue matrix and eigenvector matrix, respectively,

$$(3.10) \qquad \begin{bmatrix} I_m + \Omega_m & 0 & 0 \\ 0 & I_{n-2m} & 0 \\ 0 & 0 & I_m - \Omega_m \end{bmatrix} \text{ and } \begin{bmatrix} I_m & 0 & I_m \\ 0 & I_{n-2m} & 0 \\ I_m & 0 & -I_m \end{bmatrix},$$

where some degrees of freedom have not been expressed explicitly. Clearly this decomposition remains correct if some of the ω_i are multiple eigenvalues or $\omega_i = 0$ for some i. However, additional degrees of freedom arise for the eigenvectors in such cases. Finally, we find the singular values by taking the (positive) square roots of

the eigenvalues $1 \pm \omega_i$. From the definition of Ω we see that the largest and smallest singular value are given, respectively, by $1 + \omega_1$ and $1 - \omega_1$, which proves (3.8). The case $n - m < m$ can be proved analogously. $\quad\square$

Lemma 3.6 provides a general result about the eigenvector matrix of a projection. The proof also gives the right singular vectors (the eigenvectors in (3.10)), and the left singular vectors can be obtained by multiplication with $[U_1, U_2]$. Using Lemma 3.6, we can simplify the bound on the eigenvalues of $\mathcal{B}(S)$.

COROLLARY 3.7. *In the notation of Theorem* 3.5, *for each eigenvalue* λ_S *of* $\mathcal{B}(S)$ *there is an eigenvalue* λ *of* $\mathcal{B}(0)$ *such that*

$$(3.11) \qquad |\lambda_S - \lambda| \leq c_S \left(\frac{1 + \omega_1}{1 - \omega_1} \right)^{1/2} \|S\|,$$

where ω_1 *is the largest singular value of* $U_1^T U_2$.

In particular, if $\omega_1 = 1 - \varepsilon$, then (3.11) becomes

$$(3.12) \quad |\lambda_S - \lambda| \leq c_S \left(2\varepsilon^{-1} - 1 \right)^{1/2} \|S\| \approx \sqrt{2} \, c_S \, \varepsilon^{-1/2} \, \|S\| \approx 2.075 \, \varepsilon^{-1/2} \, \|S\|.$$

Hence, if the angles between $\mathrm{Null}(NM)$ and $\mathrm{Range}(NM)$ are not too small, the eigenvalue perturbation depends essentially on $\|S\|$. This shows that, unless the minimal angle between $\mathrm{Range}(AU_1)$ and $\mathrm{Range}(B^T)$ is very small (see the next section), the main concern in deriving good preconditioners is to find a good splitting $A = D - E$. This is a well-developed research area [15, 34]. At the end of the next section we briefly discuss what to expect for NM and the ω_i for various classes of problems.

4. An efficient implementation of the constraint preconditioner. We now derive a smaller system whose solution leads to the solution of the overall system. We extend the so-called constraint preconditioner for the symmetric case [17, 24, 25, 26] to the general case discussed here, and we derive an efficient method of solution.

Rather than apply the constraint preconditioner directly, we derive it from the preconditioned system (2.6) to emphasize the relation between the two. Consider the splitting (3.6) derived from (2.6) and the resulting system of linear equations,

$$(4.1) \qquad \mathcal{B}(0) \left[\begin{array}{c} x \\ y \end{array} \right] = \left[\begin{array}{cc} S & 0 \\ 0 & 0 \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right] + \left[\begin{array}{c} f \\ g \end{array} \right].$$

Multiplying both sides from the left by $\mathcal{B}(0)^{-1}$ (see (3.1)) leads to the fixed point iteration

$$(4.2) \qquad \left[\begin{array}{c} x_{k+1} \\ y_{k+1} \end{array} \right] = \left[\begin{array}{c} (I_n - NM)Sx_k \\ MSx_k \end{array} \right] + \left[\begin{array}{c} \hat{f} \\ \hat{g} \end{array} \right],$$

where $[\hat{f}^T, \hat{g}^T]^T \equiv \mathcal{B}(0)^{-1}[f^T, g^T]^T$. Since the right-hand side depends only on x_k, the convergence of (4.2) depends only on the iteration

$$(4.3) \qquad x_{k+1} = (I_n - NM)Sx_k + \hat{f} \equiv Fx_k + \hat{f}.$$

The observation that a splitting of (1.1) based on the constraint preconditioner (4.5) leads to a fixed point iteration that does not depend on y_k is also made in [2], but no consequences are mentioned. However, in the context of a multigrid algorithm, the fixed point iteration (4.3) is mentioned in [4] for the special case that A is the discrete Laplacian and $D = \alpha I_n$. Note that y_{k+1} in (4.2) is available essentially for free. The

iteration (4.3) converges if and only if the spectral radius of $F = (I_n - NM)S$ satisfies $\rho(F) < 1$.

A fixed point x of iteration (4.3) satisfies $x = Fx + \hat{f}$, or, equivalently,

$$(4.4) \qquad\qquad Rx = \hat{f}, \quad R \equiv I_n - F.$$

This is called the *related system* for the fixed point iteration [15], and we can solve (4.4) by a Krylov subspace method as an alternative to solving (2.6).

We now turn to the relation between the related system and a constraint preconditioner for the general problem (1.1). In our notation this preconditioner is given by

$$(4.5) \quad \begin{bmatrix} D & B^T \\ C & 0 \end{bmatrix}^{-1}$$
$$= \begin{bmatrix} D^{-1} - D^{-1}B^T (CD^{-1}B^T)^{-1} CD^{-1} & D^{-1}B^T (CD^{-1}B^T)^{-1} \\ (CD^{-1}B^T)^{-1} CD^{-1} & -(CD^{-1}B^T)^{-1} \end{bmatrix}.$$

Preconditioning (1.1) from the left by (4.5) yields

$$(4.6) \qquad \begin{bmatrix} D & B^T \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} A & B^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} D & B^T \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tilde{f} \\ \tilde{g} \end{bmatrix}$$

$$(4.7) \qquad \Leftrightarrow \begin{bmatrix} I_n - (I_n - NM)S & 0 \\ -MS & I_m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \hat{f} \\ \hat{g} \end{bmatrix},$$

where, as for the left block-diagonal preconditioned matrix $\mathcal{B}(S)$ given in (2.3), we define $N \equiv D^{-1}B^T$, $M \equiv (CD^{-1}B^T)^{-1} C$, and $S \equiv D^{-1}E$.

It turns out that in case of left (constraint and block-diagonal) preconditioning, the (1,1)-block of the matrix in (4.7) is precisely the related system matrix R. Hence, solving (4.4) and computing y from (4.2) corresponds to solving the (1,1)-block of (4.7) separately and again computing y at the end. Solving the related system rather than the whole system (4.7) has several advantages that are pointed out in section 5.

Constraint preconditioners, mainly for Hermitian matrices $\mathcal{A}$, have been discussed in many places; most useful for our discussion are [3, 4, 12, 13, 17, 21, 24, 25, 26]. In [3, 4, 13, 17, 21, 24, 25, 26] the matrix is Hermitian, and in [3, 4, 21, 24, 25, 26] the (1,1)-block of the matrix either is positive definite or made positive definite by transforming the problem. In [17] the (1,1)-block may be indefinite, but it must be nonsingular. In [12] the (1,1)-block may be nonsymmetric, but $B = C$ (in our notation) must hold. In all cases the constraint preconditioner itself is symmetric with a positive definite (1,1)-block. In fact, in [21] the (1,1)-block is diagonal, and in [4, 24, 25, 26] it is the (scaled) identity matrix. In [24, 25, 26] the authors show that a Krylov subspace method for the right preconditioned system using a constraint preconditioner and an appropriate starting guess leads to iterates that satisfy the constraints exactly, which is important for the particular application. For this purpose, constraint preconditioners have been used in optimization for some time; see the references in [13]. This feature is also used in [21] but not elaborated. We now show that this important property also holds for our efficient implementation for the general case.

THEOREM 4.1. *Take $x_0 = \hat{f}$ as an initial guess for* (4.4) *derived from left block-diagonal preconditioning. Then every iterate x_k for $k \geq 0$ computed by a Krylov subspace method will satisfy the constraint $Cx_k = \tilde{g}$ exactly.*

Proof. In case of left block-diagonal preconditioning we have

$$(4.8) \qquad \hat{f} \; = \; (I_n - NM)D^{-1}\tilde{f} + N(CD^{-1}B^T)^{-1}\tilde{g},$$

where $M = (CD^{-1}B^T)^{-1}C$ and $N = D^{-1}B^T$. Elementary computations show that $Cx_0 = C\hat{f} = \tilde{g}$ and $CR = C$, so that

$$CR^i r_0 \; = \; Cr_0 \; = \; C(\hat{f} - R\hat{f}) \; = \; 0 \qquad \text{for} \ \ i = 0, 1, \ldots.$$

A Krylov subspace method applied to a linear system with R computes iterates of the form $x_k = x_0 + \sum_{i=0}^{k-1} \alpha_i R^i r_0$, and hence $Cx_k = \tilde{g}$. $\quad\square$

If we know a better starting guess than the one in Theorem 4.1, we can easily exploit that when solving the related system (4.4). Furthermore, if we want to satisfy the constraints, we proceed according to the following theorem.

THEOREM 4.2. *Let x_0 be any initial guess for* (4.4) *obtained from left block-diagonal preconditioning, and compute x_1 following* (4.3). *Then every subsequent iterate x_{k+1} for $k \geq 1$ computed by a Krylov subspace method for* (4.4) *with initial guess x_1 will satisfy the constraint $Cx_{k+1} = \tilde{g}$ exactly.*

Proof. Computing x_1 following (4.3) means that $x_1 = Fx_0 + \hat{f}$, where $\hat{f}$ is given as in (4.8). Then the subsequent Krylov subspace method iterates are of the form $x_{k+1} = x_1 + \sum_{i=0}^{k-1} \alpha_i R^i r_1$, for $k = 1, 2, \ldots$. Now $CF = 0$, and, as in the proof of Theorem 4.1, $CR^i = C$ for $i \geq 0$, and $C\hat{f} = \tilde{g}$. Therefore

$$Cx_{k+1} \; = \; Cx_1 + \sum_{i=0}^{k-1} \alpha_i CR^i r_1 \; = \; CFx_0 + C\hat{f} + \sum_{i=0}^{k-1} \alpha_i C(x_1 - Rx_1) \; = \; \tilde{g},$$

which completes the proof. $\quad\square$

In a nutshell, when solving the related system (4.4) derived from left block-diagonal preconditioning by any Krylov subspace method, then the iterates x_k satisfy the constraints in every step when either $x_0 = \hat{f}$ (cf. Theorem 4.1) or x_0 is arbitrary and one "preprocessing" step of the fixed point iteration is performed (cf. Theorem 4.2). These properties are important for problems where the constraints must be satisfied exactly, even if the overall accuracy is allowed to be lax. This often holds for problems involving discretized conservation laws. The failure to satisfy the constraints may lead to instability and/or nonphysical solutions. In such cases, this preconditioned iteration allows the solution of x to low accuracy in a few iterations. This may yield significant computational savings.

Remark 4.3. In the discussion above we compared left block-diagonal and constraint preconditioning. In case of *right preconditioning*, the constraint preconditioner leads to a system matrix of the form

$$(4.9) \qquad \begin{bmatrix} I_n - S\,(I_n - NM) & -SN \\ 0 & I_m \end{bmatrix},$$

where, as for the right block-diagonal preconditioned matrix $\mathcal{B}(S)$, we define $N \equiv B^T\,(CD^{-1}B^T)^{-1}$, $M \equiv C\,D^{-1}$, and $S = ED^{-1}$. Hence, unlike for left preconditioning, the related system matrix R obtained from right block-diagonal preconditioning (which is still of the form $R = I_n - (I_n - NM)\,S$) is generally not equal to the (1,1)-block of (4.9). In addition, results similar to Theorem 4.1 and 4.2 do *not* hold for R derived from right block-diagonal preconditioning. When satisfying the constraints is

an important issue, we therefore recommend to always use R from left block-diagonal preconditioning.

The next important step is to bound the location of the eigenvalues of the related system matrix $R = I_n - F$ (derived either from left block-diagonal or right block-diagonal preconditioning). As shown above, in case R is derived from left block-diagonal preconditioning, this matrix is equal to the (1,1)-block of the matrix obtained from left constraint preconditioning. Assuming that $\mathcal{A}$ is symmetric, the authors of [17] show that the whole preconditioned matrix corresponding to (4.7) has $2m$ eigenvalues equal to 1. This also holds for the general case: It is clear from (4.7) that the matrix has m eigenvalues equal to 1 and the n eigenvalues of the related system matrix. Since $F \in \mathbb{R}^{n \times n}$ and $\dim(\mathrm{Range}(F)) = n - m$, we have $\dim(\mathrm{Null}(F)) \geq m$. Hence, R has (at least) an additional m eigenvalues equal to 1 corresponding to the basis vectors for $\mathrm{Null}(F)$.

For the remaining eigenvalues of R none of the approaches used in [12, 17, 21, 24, 25, 26] is applicable to our general case, as they all require D to be symmetric and $B = C$. Because of the relation $R = I_n - F$, each bound on $\rho(F)$ will simultaneously give us a bound on the distance of the eigenvalues of R from 1. Since the projection $I_n - NM$ satisfies $(I_n - NM)U_1 = U_1$ and $(I_n - NM)U_2 = 0$, we have

$$(4.10) \qquad F \;=\; (I_n - NM)\, S \;=\; [U_1, U_2] \begin{bmatrix} I_{n-m} & 0 \\ 0 & 0 \end{bmatrix} [U_1, U_2]^{-1} S.$$

THEOREM 4.4. *The spectral radius of the fixed point iteration matrix F in* (4.3) *and the eigenvalues λ_R of the related system matrix R in* (4.4) *satisfy*

$$(4.11) \qquad \left. \begin{array}{c} \rho(F) \\ |1 - \lambda_R| \end{array} \right\} \;\leq\; \frac{\|S\|}{(1 - \omega_1^2)^{1/2}},$$

where ω_1 is the largest singular value of $U_1^T U_2$.

Proof. Let $Fz = \lambda z$. Then $|z| \leq \|F\| \leq \|I - NM\|\|S\|$. The minimum principal angle between $\mathrm{Range}(NM)$ and $\mathrm{Null}(NM)$ is $\theta_1 = \arccos(\omega_1)$, and following [22, section 5.15], we get

$$(4.12) \qquad \|I - NM\| = \sin(\theta_1)^{-1} = (1 - \omega_1^2)^{-1/2}. \qquad \square$$

Similar to (3.11) we can estimate the bound (4.11) for $\omega_1 = 1 - \varepsilon \approx 1$ as follows:

$$(4.13) \qquad \left. \begin{array}{c} \rho(F) \\ |1 - \lambda_R| \end{array} \right\} \;\leq\; \frac{\|S\|}{(-\varepsilon^2 + 2\varepsilon)^{1/2}} \;\approx\; (2\varepsilon)^{-1/2}\,\|S\|.$$

Again, if the angles between $\mathrm{Range}(NM)$ and $\mathrm{Null}(NM)$ are not too small, the clustering of the eigenvalues near 1 depends essentially on $\|S\|$. This shows that, unless the minimal angle between $\mathrm{Range}(AU_1)$ and $\mathrm{Range}(B^T)$ is very small (see below), we are mainly concerned with good splittings of the (1,1)-block of the original matrix. This is emphasized further by the following invariance property of the preconditioned matrices. Suppose that we scale $\mathcal{A}$ (1.1) as follows:

$$(4.14) \qquad \tilde{\mathcal{A}} \equiv \begin{bmatrix} I_n & 0 \\ 0 & R_1 \end{bmatrix} \begin{bmatrix} A & B^T \\ C & 0 \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & R_2 \end{bmatrix} \equiv \begin{bmatrix} A & B^T R_2 \\ R_1 C & 0 \end{bmatrix},$$

where $R_1, R_2 \in \mathbb{R}^{m \times m}$ are both invertible. After splitting $A = D - E$, the corresponding preconditioner is given by

$$(4.15) \qquad \tilde{\mathcal{P}}(D) \equiv \begin{bmatrix} D^{-1} & 0 \\ 0 & (R_1 C D^{-1} B^T R_2)^{-1} \end{bmatrix}.$$

Left preconditioning yields

$$(4.16) \quad \tilde{\mathcal{P}}(D)\tilde{\mathcal{A}} \equiv \left[\begin{array}{cc} I_n - S & \tilde{N} \\ \tilde{M} & 0 \end{array} \right] = \left[\begin{array}{cc} I_n & 0 \\ 0 & R_2^{-1} \end{array} \right] \left[\begin{array}{cc} I_n - S & N \\ M & 0 \end{array} \right] \left[\begin{array}{cc} I_n & 0 \\ 0 & R_2 \end{array} \right],$$

with $S = D^{-1}E$ as before, $\tilde{N} = D^{-1}B^T R_2$, and $\tilde{M} = R_2^{-1}(CD^{-1}B^T)^{-1}C$. Thus, $\tilde{M}\tilde{N} = MN = I_m$, and $\tilde{N}\tilde{M} = NM$. This shows the following theorem.

THEOREM 4.5. *The left preconditioned matrix* $\tilde{\mathcal{P}}(D)\tilde{\mathcal{A}}$ *as given in* (4.16) *has the same eigenvalues as* $\mathcal{B}(S)$ *and also the same eigenvalue bounds relative to* $\mathcal{B}(0)$ *(cf. Theorem* 3.5 *and Corollary* 3.7*). Furthermore, it leads to the same fixed point iteration matrix* $F = (I_n - NM)S$ *and related system matrix* $R = I_n - F$ *as* $\mathcal{B}(S)$.

In short, this theorem shows that, given A and the splitting $A = D - E$, only Range(B^T) and Null(C) matter.

Remark 4.6. We have shown that the related system (4.4) derived from left preconditioning, an efficient form of using the left constraint preconditioner (4.7), can be derived from the left block-diagonal preconditioner. Furthermore, we have provided bounds on the clustering of the eigenvalues, defined in the same parameters, for all preconditioned systems. Hence, these bounds are easy to compare. We see that the related system/constraint preconditioner always leads to bounds on the clustering of eigenvalues that are better than those for the block-diagonal preconditioner (with the same splitting), especially for the extreme case when $\omega_1 \to 1$.

Next, we provide some perspective for the eigenvalue bounds derived here and in the previous section. If $B = C$ and D is symmetric positive definite, we can compute a Cholesky factorization of the preconditioner (2.2) with the same block structure, $\mathcal{P}(D)^{-1} = LL^T$. In that case, we can use symmetric preconditioning and solve for $L^{-1}\mathcal{A}L^{-T}$. It follows that NM is an orthogonal projection and $\omega_1 = 0$. For example, if our matrices and splitting $A = D - E$ satisfy the requirements in [17], their approach yields real eigenvalues in the interval $[1 - \rho(S), 1 + \rho(S)]$, where $\rho(S)$ denotes the spectral radius of S. If we assume symmetric preconditioning, then S is symmetric and $(I_n - NM)S$ is symmetric over Range($I_n - NM$). Hence, we obtain the exact same result. If we use left instead of symmetric preconditioning, the range of NM is D-orthogonal to the nullspace of NM, and the condition number of D determines ω_1; in particular, $\omega_1 \approx 1 - 2/\kappa(D)$ for $\kappa(D)$ large. We also get a similar result following the approach in [12], taking for D the (positive definite) symmetric part of A. In this case, the eigenvalues lie on the segment $[1 - i\rho(S), 1 + i\rho(S)]$, parallel to the imaginary axis. Note that this segment is included in the bound given in Theorem 4.4; see also [7, Theorem 5.3].

Finally, we consider the general case (1.1) and the associated constraint preconditioner,

$$(4.17) \qquad\qquad \mathcal{P}_c \equiv \left[\begin{array}{cc} D & B^T \\ C & 0 \end{array} \right].$$

The spaces Range(U_1) = Null(C) and Range(U_2) = Range($D^{-1}B^T$) and the value ω_1, which measures the smallest canonical angle between these spaces, play a fundamental role for the constraint preconditioner, and hence for the related system (4.4). Note that these spaces determine the projection NM. Since D is invertible, $\mathcal{P}_c$ is nonsingular if and only if $[U_1 \, U_2]$ is nonsingular. Furthermore, let $U_2 R_2 = D^{-1}B^T$, and let κ denote the spectral condition number. Then it is easy to see that $\kappa([U_1 \, U_2 R_2]) \geq \kappa([U_1 \, U_2])$. Hence, ω_1 determines an inherent lower bound on the conditioning of

$\mathcal{P}_c$. In this respect, consider the ideal case in terms of conditioning for $\mathcal{P}_c$, when D, B^T, and C^T all have orthonormal columns. From the norms of the blocks we have $1 \le \sigma_{\max}(\mathcal{P}_c) \le 3$, and we can show with some tedious algebra that for $\omega_1 \to 1$, $\sigma_{\min}(\mathcal{P}_c) = \frac{1}{2}(1 - \omega_1^2)^{1/2}$. Thus, for this special matrix $\mathcal{P}_c$ we have

$$(4.18) \qquad \frac{2}{(1 - \omega_1^2)^{1/2}} \le \kappa(\mathcal{P}_c) \le \frac{6}{(1 - \omega_1^2)^{1/2}}.$$

Of course, if $D \approx A$, ω_1 plays a similar role for $\mathcal{A}$ (approximately). Thus, ω_1 cannot be very close to 1, unless $\mathcal{P}_c$ and typically $\mathcal{A}$ are very ill conditioned. Moreover, depending on A, D, B^T, and C, $\kappa(\mathcal{P}_c)$ and $\kappa(\mathcal{A})$ may be much larger than the upper bound in (4.18).

A disadvantage of the methods discussed is that they require the inverse of the Schur-complement–type matrix $CD^{-1}B^T$. In many cases, though not always, this is expensive to compute. In the block-diagonal preconditioner one can easily replace $(CD^{-1}B^T)^{-1}$ by an approximation with minor effects on the convergence. See [7] for a discussion of eigenvalue bounds in the case of Stokes and Navier–Stokes equations. However, this simple strategy does not work for the related system. Assume we precondition $\mathcal{A}$ from the left by the block-diagonal matrix $\tilde{P} = \operatorname{diag}(D^{-1}, \tilde{S}_D^{-1})$, where $\tilde{S}_D^{-1}$ is an approximation to $S_D^{-1} \equiv (CD^{-1}B^T)^{-1}$. Then $\tilde{S}_D^{-1}S_D$ is invertible and

$$\tilde{P}\mathcal{A} \;=\; \left[\begin{array}{cc} I_n - S & D^{-1}B^T \\ \tilde{S}_D^{-1}C & 0 \end{array} \right] \;=\; \left[\begin{array}{cc} I_n & 0 \\ 0 & \tilde{S}_D^{-1}S_D \end{array} \right] \left[\begin{array}{cc} I_n - S & N \\ M & 0 \end{array} \right].$$

Theorem 4.5 shows that the related system derived from this matrix is again $I_n - (I_n - NM)S$, requiring $(CD^{-1}B^T)^{-1}$. Thus, for a related system with an approximate Schur complement we need an alternative. We will discuss this in more detail in [19].

5. GMRES for the system with $\mathcal{B}(S)$ versus GMRES for the related system. We summarize the results from the previous sections to compare two approaches for solving the linear system (1.1):

1. apply GMRES to (2.6), i.e., to a system with the matrix $\mathcal{B}(S)$.
2. apply GMRES to (4.4), i.e., to a system with the matrix R, or, equivalently, to the (1,1)-block of (4.7) and the associated right-hand side.

In general, the convergence of GMRES depends on the eigenvalues and eigenvectors of the given system matrix and their relation to the initial residual. As we have made practically no assumptions on $\mathcal{A}$ or on the right-hand side of (1.1), this is difficult to analyze. Therefore, we look only at the eigenvalue clustering of $\mathcal{B}(S)$ and R as an indication of the convergence behavior of GMRES for (2.6) and (4.4).

The following considerations show why in order to solve (1.1) it is often more efficient to apply GMRES to the related system (4.4) than to the preconditioned system (2.6):

1. The iterates x_k of GMRES applied to the related system (4.4) derived from left block-diagonal preconditioning with the initial guess following either Theorem 4.1 or Theorem 4.2 satisfy the constraints $Cx_k = \tilde{g}$ in (1.1) exactly; this is not the case for the iterates from GMRES applied to (2.6).
2. This property of using the related system leads to further advantages if we scale the constraints to improve convergence and use an inexact Newton (iterative) solver; see sections 6 and 7.
3. The size of the related system matrix R is $n \times n$, while the size of $\mathcal{B}(S)$ is $(n + m) \times (n + m)$. This size advantage of R is particularly important for

methods like GMRES that have to store and orthogonalize many vectors. We note that the costs of computing matrix-vector products with R and $\mathcal{B}(S)$ are similar. In both cases we have to perform one multiplication each with M, N, and S.

4. The related system matrix R has all eigenvalues clustered around 1 and so is definite (positive real) in the case of good clustering. The matrix $\mathcal{B}(S)$ remains indefinite and has three clusters of eigenvalues in the case of good clustering.

5. The *center* of the (only) eigenvalue cluster of R is 1. The *center* of the eigenvalue cluster of $\mathcal{B}(S)$ which is closest to the origin is $(1 - \sqrt{5})/2 \approx -0.6$, and thus is closer to zero.

6. For $\omega_1 = 1 - \varepsilon$, the bound (4.13) for the eigenvalues of R is almost a factor 3 smaller than the bound (3.12) for the eigenvalues of $\mathcal{B}(S)$.

Although the latter three advantages appear to be small, we think they are important. In many cases we will have a preconditioner that is effective but does not give very tight clustering. In such a case, the convergence of GMRES applied to the related system (4.4) may be significantly better than that of GMRES applied to the preconditioned system (2.6). Note that the preconditioned iterations may be quite expensive. Furthermore, if a fair number of iterations of GMRES is required, then the reduction of the problem size is even more important. Moreover, satisfying the constraints even when the solution is not highly accurate is important for many applications.

Finally, we note that the comparisons (2) and (6) are new even in the context of the symmetric case. Moreover, it appears that eigenvalue bounds for the two types of preconditioners in terms of the same parameters, which make comparison (6) possible, have not been derived before. In the symmetric case, if D is symmetric positive definite, the difference between the bounds reduces to a factor of about 1.5, because symmetric preconditioning will make all $\omega_i = 0$.

Our numerical examples in section 7 show that using the related system (4.4) instead of the preconditioned system (2.6) can lead to significant savings in solving (1.1).

6. Surface parameterization as a constrained optimization problem and scaling nonlinear constraints. The application that we use to illustrate our theoretical results is a constrained optimization problem that arises in mesh-flattening [28, 29], the most expensive step in surface parameterization. The latter is of considerable interest in many areas [29], such as generating a surface mesh for three-dimensional finite element meshing, and texture mapping in graphics [30].

The basic idea of mesh-flattening is to compute a flat triangulation that is isomorphic to a given faceted surface (patch) with minimal angular deformation. The algorithm we briefly describe next computes mesh (b) from mesh (a) in Figures 6.1 and 6.2. Meshes (c) and (d) in Figure 6.1 illustrate the generation of a better (and in this case coarser) mesh using the flat triangulation; for details see [29].

To minimize angular deformation, we wish to minimize the function

$$(6.1) \qquad \mathcal{Q}(\alpha) = \sum_{i=1}^{\#\text{faces}} \sum_{j=1}^{3} (\alpha_i^j - \phi_i^j)^2 w_i^j,$$

where α_i^j is the jth angle in face i in the flat mesh, ϕ_i^j is the *optimal angle* for α_i^j, and w_i^j is a weight. Typically $w_i^j = (\phi_i^j)^{-2}$, minimizing the relative deformation of angles.
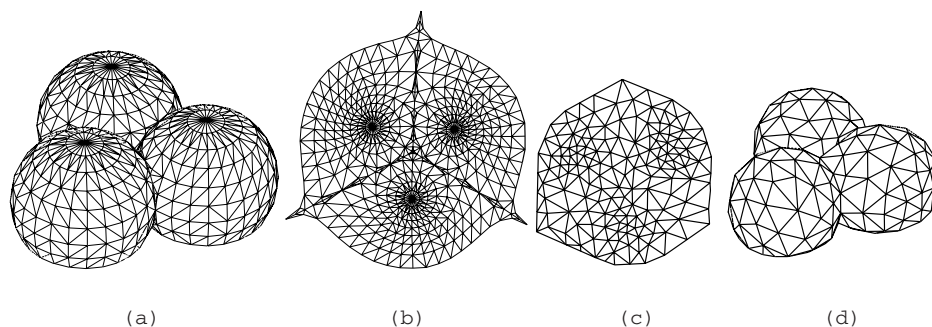
FIG. 6.1. *Remeshing the three balls. From left to right are* (a) *the original surface mesh,* (b) *the computed flat mesh,* (c) *the coarser remeshing in the plane, and* (d) *the new mesh mapped back to the three-dimensional surface. Note the improved quality (no very small angles) of the faces in the coarser mesh. In particular, compare the regions near the poles in* (a) *and* (d).
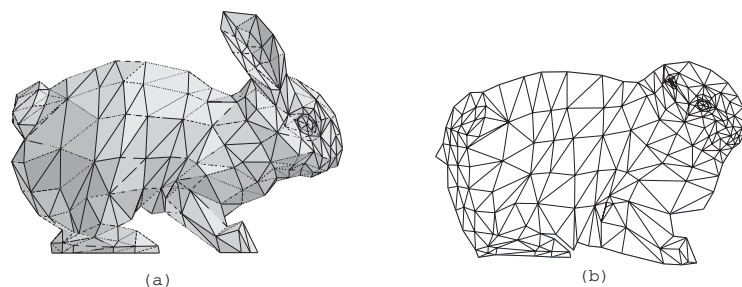

FIG. 6.2. *Flattening the half-rabbit mesh.* (a) *The original surface mesh and* (b) *the computed flat mesh.*

The optimal angles at interior nodes are derived from the inevitable local deformation that results from flattening a nonsmooth surface.

The angles in the flat mesh need to satisfy four classes of constraints. We denote by $\mathcal{G}^{(i)}(\alpha)$ the row vector of all constraints in class $i = 1, 2, 3, 4$. The first class of constraints is that all angles must remain positive. We handle this constraint algorithmically [29], and so will not discuss it below. The second class of constraints is that the angles inside each triangle sum to π. The third class of constraints is that the angles at each interior node sum to 2π. These constraints are linear. Finally, the fourth class of constraints is that neighboring faces must agree on the size of the shared edge. This leads to one nonlinear constraint for each interior node of the form

$$(6.2) \qquad \Pi_i \sin(\alpha_i^{j(k)+1}) - \Pi_i \sin(\alpha_i^{j(k)-1}) = 0,$$

where $\alpha_i^{j(k)}$ indicates the angle in face i at the interior node N_k, and i runs over the faces containing node N_k. To demonstrate the effects of scaling the nonlinear constraints we scale the constraints $\mathcal{G}^{(4)}(\alpha)$ by η. For convenience (see below), we also scale the constraints $\mathcal{G}^{(3)}(\alpha)$.

This leads to the constrained minimization problem

$$\begin{aligned} &\min \ \mathcal{Q}(\alpha) \quad \text{subject to} \\ (6.3) \qquad &\mathcal{G}(\alpha) \equiv [\mathcal{G}^{(2)}(\alpha), \ \eta\mathcal{G}^{(3)}(\alpha), \ \eta\mathcal{G}^{(4)}(\alpha)]^T = 0. \end{aligned}$$

Applying the Lagrange multiplier formulation, we use Newton's method to find a
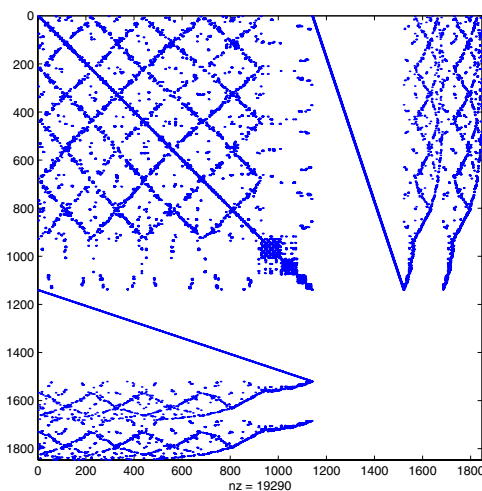
FIG. 6.3. *Structure of a system matrix* $\mathcal{A}_{\eta,l}$ *derived from the half-rabbit model.*

critical point of the Lagrangian

$$(6.4) \qquad \mathcal{L}(\alpha, \lambda) \equiv \mathcal{Q}(\alpha) + \lambda^T \mathcal{G}(\alpha),$$

where λ is the vector of Lagrange multipliers. In each Newton step we must solve the system of equations

$$(6.5) \qquad \mathcal{A}_{\eta,l} \left[\begin{array}{c} x \\ y \end{array} \right] \equiv \left[\begin{array}{cc|c} Q + \eta H_l & J_1^T & \eta J_{2,l}^T \\ J_1 & 0 & 0 \\ \hline \eta J_{2,l} & 0 & 0 \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right] = \left[\begin{array}{c} f_\eta \\ g_\eta \end{array} \right],$$

where

$$Q = \nabla_\alpha^2 \mathcal{Q}(\alpha), \quad H_l = (\lambda^{(4)})^T \nabla_\alpha^2 \mathcal{G}^{(4)}(\alpha), \quad J_1 = \nabla_\alpha \mathcal{G}^{(2)}(\alpha), \quad J_{2,l} = \nabla_\alpha [\mathcal{G}^{(3)}(\alpha) \, \mathcal{G}^{(4)}(\alpha)],$$

and

$$(6.6) \qquad \left[\frac{f_\eta}{g_\eta} \right] = \left[\begin{array}{c} -\nabla_\alpha \mathcal{Q} - \lambda^T \nabla_\alpha [\mathcal{G}^{(2)} \ \eta \mathcal{G}^{(3)} \ \eta \mathcal{G}^{(4)}] \\ -\mathcal{G}^{(2)} \\ \hline -\eta \mathcal{G}^{(3)} \\ -\eta \mathcal{G}^{(4)} \end{array} \right].$$

In these equations, only the block matrices with subscript l change from one Newton step to the next. The structure of an example matrix $\mathcal{A}_{\eta,l}$ is shown in Figure 6.3. To define the two-by-two block form of $\mathcal{A}_{\eta,l}$ we take

$$(6.7) \qquad A_l = \left[\begin{array}{cc} Q + \eta H_l & J_1^T \\ J_1 & 0 \end{array} \right] \quad \text{and} \quad B_l = C_l = [\eta J_{2,l} \ 0].$$

We consider the splitting $A_l = D - \eta E_l$ (cf. (2.1)), taking

$$(6.8) \qquad D = \left[\begin{array}{cc} Q & J_1^T \\ J_1 & 0 \end{array} \right],$$

$$(6.9) \qquad E_l = \left[\begin{array}{cc} H_l & 0 \\ 0 & 0 \end{array} \right].$$

For this choice of D, D^{-1} is known explicitly. A formula for D^{-1} and more details on the structure of $\mathcal{A}_l$ are given in [20]. Let $S_l = D^{-1}E_l$, $N_l = D^{-1}B_l^T$, $M_l = \left(B_l D^{-1}B_l^T\right)^{-1}B_l$, and $F_l = (I - N_l M_l)S_l$, all corresponding to $\eta = 1$. We get the (left) preconditioned system

$$(6.10) \qquad P_\eta(D)\mathcal{A}_{\eta,l} = \left[\begin{array}{c|c} I - \eta S_l & \eta N_l \\ \hline \eta^{-1}M_l & 0 \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} D^{-1}f_\eta \\ \hline \eta^{-1}\left(B_l D^{-1}B_l^T\right)^{-1}g \end{array}\right].$$

Now following Theorem 4.5 with $R_1 = \eta I$ and $R_2 = \eta^{-1}I$, the eigenvalue bounds of (6.10) are the same as the eigenvalue bounds for the matrix

$$(6.11) \qquad\qquad \left[\begin{array}{c|c} I - \eta S_l & N \\ \hline M & 0 \end{array}\right],$$

and the fixed point iteration matrix and related system become, respectively,

$$
\begin{aligned}
F_{\eta,l} &= \eta(I - N_l M_l)S_l = \eta F_l, \\
(6.12) \qquad R_{\eta,l}x &= \hat{f}_{\eta,l}, \quad \text{where} \quad R_{\eta,l} \equiv I - \eta F_l.
\end{aligned}
$$

Note that the right-hand side differs from that of the problem without scaling.

We see that in (6.11) and (6.12) only ηF_l and ηS_l, respectively, depend on η. This leads to the following eigenvalue bounds. For $P_\eta(D)\mathcal{A}_{\eta,l}$ we get (cf. (3.4) and (3.11))

$$(6.13) \qquad\qquad |\lambda_{\eta,S_l} - \lambda| \le \eta c_S \left\| [U_{1,l}\,U_{2,l}]^{-1}S_l[U_{1,l}, U_{2,l}] \right\|$$

$$(6.14) \qquad\qquad\qquad\qquad \le \eta c_S \left(\frac{1 + \omega_{1,l}}{1 - \omega_{1,l}}\right)^{1/2}\|S_l\|,$$

and for $F_{\eta,l}$ and $R_{\eta,l}$ we get (cf. (4.11))

$$(6.15) \qquad\qquad \left.\begin{array}{c} \rho(F_{\eta,l}) \\ |1 - \lambda_{R_{\eta,l}}| \end{array}\right\} \le \frac{\eta}{(1 - \omega_{1,l})^{1/2}}\|S_l\|,$$

where $\omega_{1,l}$ is the largest singular value of $U_1^T U_2$ derived from $I - N_l M_l$.

   Therefore we can make the eigenvalue clustering arbitrarily close!

   Obviously, there is a catch in scaling the constraints. It can be seen from (6.5) and (6.6) that for small η the nonlinear system approximates a quadratic problem with linear constraints. The stationary point of such a system is given by the solution of a linear system, and Newton's method will converge in a single iteration. Moreover, our preconditioners would be the ideal preconditioners. Thus, for small η we solve a sequence of linear systems that are close to the linear system corresponding to a quadratic problem with linear constraints. Hence, the convergence of Newton's method may slow down. Although the left preconditioned system and the related system lead to the same solution, this convergence behavior is most obvious for the related system. We see from (6.12) that for small η the related system matrix gets close to the identity and hence becomes easier to solve. However, at the same time the nonlinear components of the problem are relatively small, and hence the Newton steps tend to be less effective, resulting in slower convergence of the Newton iteration. Nevertheless, such scaling is useful to balance the cost of solving the linear systems with the number of Newton iterations to reduce overall runtime. This is similar to tuning the time-step in time-dependent problems, where a small time-step yields a

well-conditioned problem and fast convergence but necessitates more time-steps to reach the final simulation time. Note that for the preconditioned system (6.10) the constraints get a weight inversely proportional to η.

Finally, with our choice of preconditioners the convergence of the linear systems for an optimization problem with nonlinear constraints can potentially be improved by judicious scaling. For more general problems the effect on the Newton iteration will be more complicated to assess.

7. Numerical examples. We discuss the performance of GMRES [27] for a problem arising in flattening the half-rabbit mesh; cf. Figure 6.2. Typically, our mesh-flattening algorithm takes only five to ten Newton steps to converge. For our experiments reported here we picked the Jacobian $\mathcal{A}_{\eta,l}$, for which the unpreconditioned linear system (6.5) required the most GMRES steps. For the half-rabbit model this was the fourth Jacobian. To eliminate a possible correlation between the matrix $\mathcal{A}_{\eta,l}$ and the right-hand side of (6.5) as well as the initial residual, we used a random right-hand side (generated by MATLAB's `randn` function) and a zero initial guess. The matrix $\mathcal{A}_{\eta,l}$ is of order 1846 for the half-rabbit model, and the related system matrix has order 1520. The approximate Schur complement, $B_l D^{-1} B_l^T$ (of order 326), is inverted using MATLAB's Cholesky decomposition. One Cholesky decomposition is required in each Newton step. Since D^{-1} is known explicitly, the computation of this Cholesky decomposition is the most expensive part in computing the preconditioner.

Figure 7.1 shows the computed relative residual norms, $\|r_n\|/\|r_0\|$, for GMRES applied to the unpreconditioned system (6.5), the right block-diagonal preconditioned system, the left block-diagonal preconditioned system (6.10), and the related system (6.12). For the unpreconditioned system GMRES stagnates almost completely. Furthermore, the convergence for the related system is significantly better than the convergence for the left and right preconditioned systems, which confirms our theoretical results.
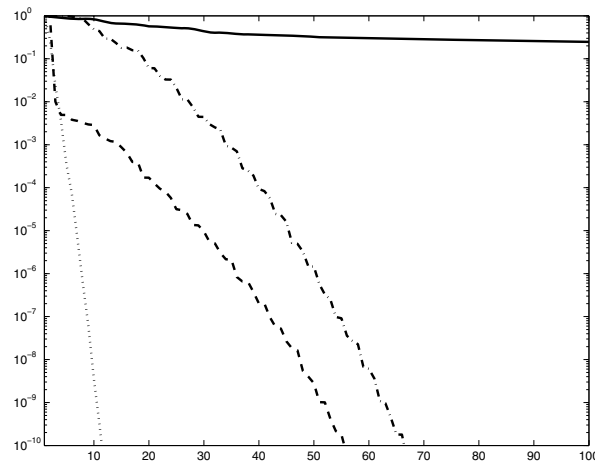

FIG. 7.1. *GMRES performance for systems derived from the half-rabbit model: unpreconditioned (solid), right preconditioned (dash-dot), left preconditioned (dashed), and related system (dots).*

The effect of scaling the constraints on the eigenvalue clustering for the matrix derived from the half-rabbit model is shown in Figures 7.2–7.4. Figure 7.2 shows why unpreconditioned GMRES performs so poorly. The eigenvalues of the original matrix
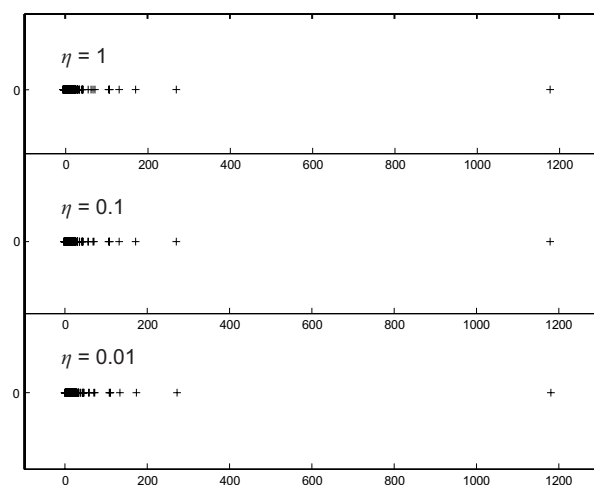
FIG. 7.2. *Eigenvalues of the original system matrix derived from the half-rabbit model with scalings $\eta = 1$, 0.1, 0.01.*
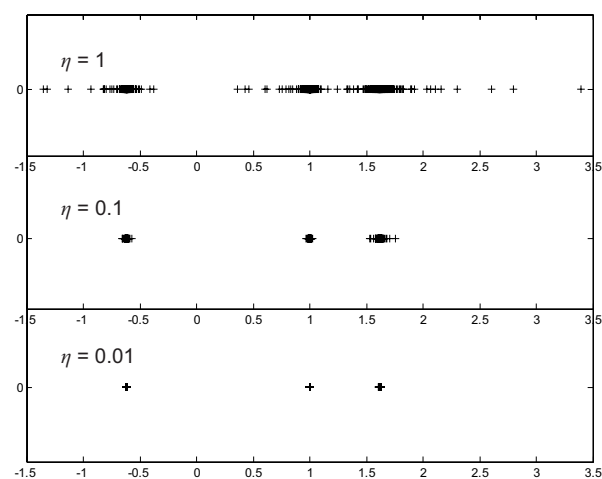

FIG. 7.3. *Eigenvalues of the left preconditioned system matrix derived from the half-rabbit model with scalings $\eta = 1$, 0.1, 0.01.*

vary by several orders of magnitude, the matrix is indefinite, and many eigenvalues are close to zero. Furthermore, scaling by $\eta = 0.1$ and 0.01 does not improve the clustering noticeably. Left preconditioning clusters the eigenvalues close to $\{1, \frac{1}{2}(1 \pm \sqrt{5})\}$; see Figure 7.3. In accordance with the bounds (6.13) and (6.14), the clustering improves when the third and fourth constraints are scaled. For the related system matrix, we see only one cluster of eigenvalues, which becomes very small when we scale by $\eta = 0.1$ and 0.01; see Figure 7.4. For $\eta = 0.01$, all eigenvalues are contained in the interval $[0.9973, 1.0029]$.

The convergence of GMRES for the unpreconditioned, the left block-diagonal preconditioned, and the corresponding related system with scaling of the third and fourth groups of constraints by $\eta = 1$ and 0.1 is shown in Figure 7.5. When using the related system, the scaling has a dramatic effect. The relative residual norm converges
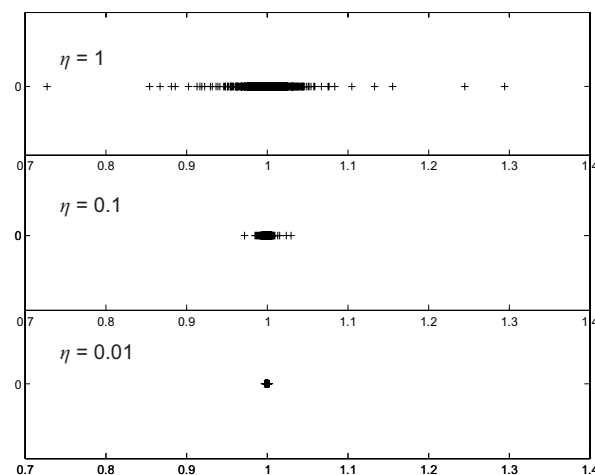
FIG. 7.4. *Eigenvalues of the related system matrix derived from the half-rabbit model with scalings $\eta = 1$, 0.1, 0.01.*
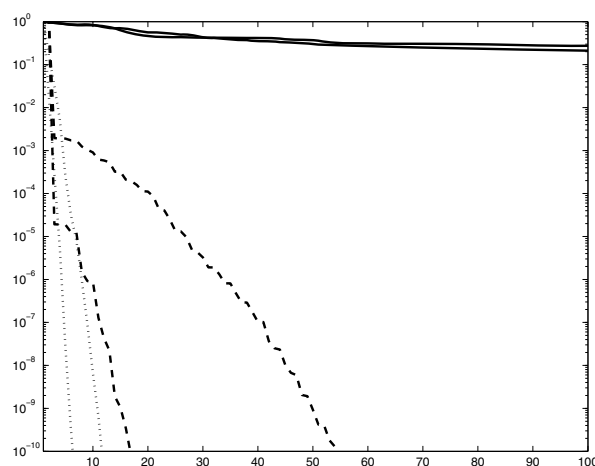

FIG. 7.5. *GMRES performance for systems derived from the half-rabbit model: unpreconditioned (solid), left preconditioned (dashed), and related system (dots), with scalings $\eta = 1$ and $\eta = 0.1$. The latter leads to significantly faster convergence for the left preconditioned and the related systems.*

to a tolerance of 10^{-10} in seven steps when the third and fourth groups of constraints are scaled by $\eta = 0.1$. The rate of convergence comes close to that obtained using the algebraically optimal preconditioner derived by Murphy, Golub, and Wathen [23]. For the scaled left preconditioned system we also see a significant improvement in the speed of convergence of GMRES. However, the scaled preconditioned matrices remain indefinite (cf. Figure 7.3), so that the problem is still more difficult for GMRES to solve than the scaled related system. This confirms that using the related system is preferable to using the left preconditioned system.

For lack of space we are here restricted to reporting on a single application only. However, essentially the same behavior can be seen for larger problems as well [5]. Finally, notice that, in principle, this convergence improvement is applicable to any optimization problem with nonlinear constraints.

8. Conclusions. We have extended two classes of preconditioners to general block two-by-two linear systems with zero (2,2)-block, including the analysis of the preconditioned systems. This required the introduction of new "tools," particularly for the analysis of oblique projections. We have developed a framework to analyze and compare block-diagonal preconditioners and constraint preconditioners, in particular the efficient implementation of constraint preconditioners introduced in this paper. So far, these preconditioners have typically been treated separately. Our analysis reveals that solving the original block two-by-two system by solving the related system is typically the best. Not only is the related system smaller in size, and not only does it have (typically) the best eigenvalue clustering, but, in addition, with the correct starting guess, the iterates of any Krylov subspace method applied to the related system (derived from left block-diagonal preconditioning) exactly satisfy the constraints imposed in the original system. With the exception of the smaller size, these conclusions also hold for constraint preconditioners for the general systems discussed here. In addition, our analysis and the invariance property from Theorem 4.5 led us to the concept of scaling nonlinear constraints in optimization problems to improve the convergence of the preconditioned linear systems arising in the solution of such problems. This scaling appears to work quite well.

Our approach is very general, as we have made practically no assumptions on the original system. Furthermore, our framework leaves a variety of choices for the user (choice of splitting, scalings, etc.). We have demonstrated the efficiency of our methods for Jacobians that arise in an optimization problem with nonlinear constraints, and we will give a more detailed numerical study in the forthcoming second part of this two-part sequence of papers [19].

Finally, we should mention that other preconditioners proposed for saddle point problems may be competitive or better for the test problem used in this paper. However, finding the best preconditioner for this problem was not the purpose of the present paper, and remains as future work. Probably, any competitive preconditioner should have the property, discussed in section 6, that it significantly improves the spectrum of the preconditioned systems by scaling the nonlinear constraints appropriately.

REFERENCES

[1] R. E. BANK, B. D. WELFERT, AND H. YSERENTANT, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.
[2] D. BRAESS, *Finite Elements*, 2nd ed., Cambridge University Press, Cambridge, UK, 2001.
[3] D. BRAESS, P. DEUFLHARD, AND K. LIPNIKOV, *A subspace cascadic multigrid method for mortar elements*, Computing, 69 (2002), pp. 205–225.
[4] D. BRAESS AND R. SARAZIN, *An efficient smoother for the Stokes problem*, Appl. Numer. Math., 23 (1997), pp. 3–19.
[5] E. DE STURLER AND J. LIESEN, *Block-Diagonal Preconditioners for Indefinite Linear Algebraic Systems. Part* I: *Theory*, Tech. Report UIUCDCS-R-2002-2279, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.
[6] N. DYN AND W. E. FERGUSON, JR., *The numerical solution of equality constrained quadratic programming problems*, Math. Comp., 41 (1983), pp. 165–170.
[7] H. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Iterative Methods for Problems in Computational Fluid Dynamics*, Proceedings of the Winter School on Iterative Methods in Scientific Computing and Applications, Chinese University of Hong Kong, 1995 (published in 1996).
[8] H. C. ELMAN, *Preconditioning for the steady-state Navier–Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.

[9] H. C. ELMAN AND G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[10] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Proceedings of the 6th Biennial Dundee Conference on Numerical Analysis, G. Watson, ed., Lecture Notes in Math. 506, Springer-Verlag, Berlin, 1976, pp. 73–89.

[11] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[12] G. H. GOLUB AND A. J. WATHEN, *An iteration for indefinite systems and its application to the Navier–Stokes equations*, SIAM J. Sci. Comput., 19 (1998), pp. 530–539.

[13] 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.

[14] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers in Appl. Math. 17, SIAM, Philadelphia, 1997.

[15] L. A. HAGEMAN AND D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.

[16] L. HEMMINGSSON-FRÄNDÉN AND A. WATHEN, *A nearly optimal preconditioner for the Navier–Stokes equations*, Numer. Linear Algebra Appl., 8 (2001), pp. 229–243.

[17] 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.

[18] P. KRZYŻANOWSKI, *On block preconditioners for nonsymmetric saddle point problems*, SIAM J. Sci. Comput., 23 (2001), pp. 157–169.

[19] J. LIESEN AND E. DE STURLER, *Block-Diagonal and Constraint Preconditioners for Nonsymmetric Indefinite Linear Systems. Part* II: *Applications*, in preparation.

[20] J. LIESEN, E. DE STURLER, A. SHEFFER, Y. AYDIN, AND C. SIEFERT, *Preconditioners for indefinite linear systems arising in surface parameterization*, in Proceedings of the 10th International Meshing Round Table, Sandia National Laboratories, Albuquerque, NM, 2001, pp. 71–81.

[21] 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.

[22] C. D. MEYER, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[23] 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.

[24] 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.

[25] I. PERUGIA, V. SIMONCINI, AND M. ARIOLI, *Linear algebra methods in a mixed approximation of magnetostatic problems*, SIAM J. Sci. Comput., 21 (1999), pp. 1085–1101.

[26] M. ROZLOŽNÍK AND V. SIMONCINI, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 368–391.

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

[28] A. SHEFFER AND E. DE STURLER, *Parameterization of CAD surfaces for meshing by triangulation flattening*, in Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, Whistler, BC, Canada, 2000, International Society of Grid Generation, Birmingham, AL, 2000, pp. 699–708.

[29] A. SHEFFER AND E. DE STURLER, *Surface parameterization for meshing using angle-based flattening*, Engineering with Computers, 17 (2001), pp. 326–337; also available online from Springer at http://link.springer.de/link/service/journals/00366/tocs/t1017003.htm.

[30] A. SHEFFER AND E. DE STURLER, *Smoothing an overlay grid to minimize linear distortion in texture mapping*, ACM Trans. Graphics, 21 (2002), pp. 874–890.

[31] D. SILVESTER, H. ELMAN, D. KAY, AND A. WATHEN, *Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow*, J. Comput. Appl. Math., 128 (2001), pp. 261–279.

[32] D. SILVESTER AND A. WATHEN, *Fast iterative solution of stabilised Stokes systems Part* II. *Using general block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.

[33] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[34] R. S. VARGA, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.

[35] A. WATHEN AND D. SILVESTER, *Fast iterative solution of stabilised Stokes systems Part* I. *Using simple diagonal preconditioners*, SIAM J. Numer. Anal., 30 (1993), pp. 630–649.