Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands

Phone: +31-53-4893400
Fax: +31-53-4893114

Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1786

A class of nonsymmetric preconditioners for saddle point problems

M.A. Botchev and G.H. Golub[1]

December, 2005

ISSN 0169-2690

[1] Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305-9025, USA

A CLASS OF NONSYMMETRIC PRECONDITIONERS FOR SADDLE POINT PROBLEMS

DEDICATED TO HENK A. VAN DER VORST ON OCCASION OF HIS 60TH BIRTHDAY

MIKE A. BOTCHEV[*] AND GENE H. GOLUB[†]

Abstract. For the iterative solution of saddle point problems, a nonsymmetric preconditioner is studied which, with respect to the upper-left block of the system matrix, can be seen as a variant of SSOR. An idealized situation where SSOR is taken with respect to the skew-symmetric part plus the diagonal part of the upper-left block is analyzed in detail. Since action of the preconditioner involves solution of a Schur complement system, an inexact form of the preconditioner can be of interest. This results in an inner-outer iterative process. Numerical experiments with solution of linearized Navier-Stokes equations demonstrate the efficiency of the new preconditioner, especially when the left-upper block is far from symmetric.

Key words. saddle point problems, iterative methods, preconditioning methods, nonsymmetric indefinite linear systems, SSOR, constraint preconditioners, skew-symmetric preconditioners, inner-outer iterations, Navier-Stokes equations

AMS subject classifications. 65F10, 65F22, 65F35, 65N22, 65K10

We are dedicating this paper to Henk van der Vorst who has made so many seminal contributions and who has been so supportive to his colleagues and our community.

1. Introduction.

We consider a nonsymmetric preconditioner for the iterative solution of the linear system

$$\mathcal{A}u = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} = b\,, \tag{1.1}$$

where $A \neq A^T \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, $C = C^T \in \mathbb{R}^{m\times m}$, and $m < n$ (often $m \ll n$). We assume that the matrix $A + A^T$ is positive definite (i.e. A is a *positive real* matrix) and C is positive semidefinite.

Linear systems of the form (1.1) arise in a number of applications including mixed finite element solution of the Navier-Stokes and the Maxwell equations and constraint optimization [24, 10, 42, 43]. In many cases A, B and C are large sparse matrices and iterative techniques are preferable for solving (1.1), especially in connection with the discretization of partial differential equations in three dimensions. Since $\mathcal{A}$ is indefinite and often ill-conditioned, preconditioning is in most cases indispensable for iterative solution of (1.1).

Let H and S be symmetric (Hermitian) and skew-symmetric (skew-Hermitian) parts of A respectively:

$$H \equiv \frac{1}{2}(A + A^T), \qquad S \equiv \frac{1}{2}(A - A^T).$$

To solve (1.1), Golub and Wathen [30] considered a basic iteration of the form

$$\mathcal{P}u^{k+1} = (\mathcal{P} - \mathcal{A})u^k + b \tag{1.2}$$

[*] Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands, mbotchev@na-net.ornl.gov. This author's research was supported in part by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (http://www.bsik-bricks.nl), theme MSV1.

[†] Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305-9025, USA, golub@sccm.stanford.edu. The work of this author was supported in part by the Department of Energy of the United States Government.

with symmetric ($P = P^T$) indefinite preconditioner

$$\mathcal{P} = \begin{bmatrix} P & B^T \\ B & -C \end{bmatrix}. \tag{1.3}$$

When A is not far from a symmetric matrix (i.e. $\|S\|/\|H\|$ is a small number), an efficient preconditioner can be obtained by taking P to be the symmetric part of A [30]. In the context of the systems stemming from the Navier-Stokes equations, $\mathcal{P}$ corresponds to a discretization of the Stokes operator and, to compute $\mathcal{P}^{-1}u$ for a given vector u, a number of robust direct and iterative techniques exists. Other choices of $P = P^T$ can also be useful [30].

However, as can be expected, performance of these symmetric preconditioners deteriorates when A is essentially nonsymmetric ($\|S\|/\|H\| \approx 1$ or larger). A need for a good nonsymmetric preconditioner and, in particular, a possibility to extend the approach of [30] to a nonsymmetric case, motivated our research.

A variety of preconditioning methods to solve (1.1) iteratively have been significantly extended within the last decade [5]. Following one possible classification [27, 4, 5], we mention block and approximate Schur complement preconditioners [18, 19, 16, 20, 17, 44, 45, 37, 38], preconditioners based on the Uzawa algorithm [25, 26, 8, 9, 59], preconditioners stemming from the classical splitting iterative schemes (where our approach may fall into) [14, 30, 31, 4], preconditioners inspired by analysis of the underlying continuous partial differential operators [35], sparse direct and approximate factorization preconditioners [13, 23, 46], the so-called null-space preconditioners [48, 1, 32], multigrid preconditioners [51], and other approaches. A few of these approaches work well in the nonsymmetric case, among them [18, 16, 44, 35, 4]. Preconditioners for systems (1.1) stemming from the Navier-Stokes equations are subject of a recent book [21].

We note that preconditioners of the form (1.3) are studied in [30, 36, 32, 11, 12, 49] and sometimes called constraint preconditioners.

For simplicity, without loss of generality, here and throughout the paper we assume that A has ones on its main diagonal, i.e.

$$\mathrm{Diag}(A) = I,$$

with $\mathrm{Diag}(A)$ and I being respectively the diagonal part of matrix A and the identity matrix. This usually can be achieved by a diagonal prescaling. In certain applications, however, one may want to avoid diagonal prescaling due to ill-conditioning of A.

In this paper we consider a nonsymmetric preconditioner which, with respect to A, can be seen as a variant of SSOR. Namely, we take P in (1.3) as

$$P := P_{\mathsf{ssor}} \equiv \frac{1}{\omega}(I + \omega L)(I + \omega U), \qquad L + U = A - I, \tag{1.4}$$

where L and U are strictly lower and upper triangular parts of A, respectively. We are not able to provide any rigorous analysis for preconditioner (1.4) and analyze an idealized situation where we take

$$P := P_{\mathsf{skew}} \equiv \frac{1}{\omega}(I + \omega L_S)(I + \omega U_S), \qquad L_S + U_S = S\,, \tag{1.5}$$

i.e. L_S and U_S are respectively lower and upper triangular parts of the skew-symmetric part of A. Evidently, when the skew-symmetric part S is large compared to the symmetric part H, P_{skew} appears to be a good approximation to P_{ssor}. Preconditioner (1.5) is thus relevant for understanding the behavior of preconditioner (1.4) when A is strongly nonsymmetric. In our analysis we use a technique similar to [39, 40, 6].

Preconditioners (1.4) and (1.5) coincide when A is a sum of the identity and a skew-symmetric matrix. The idea to consider this special situation to gain understanding in the iterative solution of nonsymmetric problems is not new [28].

Since in (1.3) the preconditioner $\mathcal{P}$ is not a product of (block) triangular matrices, an important question is how to implement action of the preconditioner, i.e. how to find $\mathcal{P}^{-1}u$ when u is given. Unlike for the symmetric Stokes preconditioner, for the preconditioners (1.4), (1.5) there are no standard solvers available. However, the system with the matrix P in our case can easily be solved and, following a straightforward approach, to compute $\mathcal{P}^{-1}u$ one needs to solve a system with the matrix $BP^{-1}B^T + C$ (the negative of the Schur complement matrix). This is an $m \times m$ matrix and in many cases, especially when $m \ll n$, solution by a direct solver would be feasible. An alternative is to apply an inexact form of the preconditioner where, for example, GMRES iteration [47] can be applied to solve the system with $BP^{-1}B^T + C$. We analyze this inexact form of our preconditioning method. Furthermore, our (limited) experience shows that this inexact preconditioner works well, leading to only a moderate increase in the number of the outer iterations as compared with the exact form.

As our numerical experiments suggest, the SSOR preconditioner (1.4) compares favorably with other preconditioning techniques, for a wide range of $\|S\|/\|H\|$, i.e. for matrices close to symmetric as well as for strongly nonsymmetric matrices.

The paper is organized as follows. In Section 2.1 analysis for the “idealized” skew preconditioner (1.3), (1.5) is given. First, we obtain general conditions to have convergence in iteration (1.2). (Convergence means that matrix $\mathcal{P}^{-1}\mathcal{A}$ has its eigenvalues on the complex plane inside the unit circle centered at the point $1 + 0i$ ($i^2 = -1$).) Then in Section 2.2 we provide bounds for ω that guarantee convergence and further discuss a possible way to optimize convergence by a suitable choice of ω. However, this optimization is usually not efficient in practice since it is based on an estimate which is not sharp. Therefore, in Section 2.3 we discuss simple ways to choose ω which work well in practice. This and subsequent sections deal both with the preconditioners (1.3), (1.4) and (1.3), (1.5). A simple model problem for which the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ can be computed analytically gives an insight into the effect of the preconditioners in Section 2.4. Inexact form of the preconditioners, where at each “outer” iteration the system with $BP^{-1}B^T + C$ is solved by an “inner” iterative process, is studied in Section 2.5. Section 2.6 addresses implementation issues and provides estimates of the computational costs. In Section 3 we present results of numerical tests. Finally, we make conclusions and give an outlook to future research in the last section.

2. Analysis of the preconditioner.

2.1. Convergence.

In this section we analyze the skew preconditioner (1.3), (1.5). Throughout this and the next subsection it is assumed that $P = P_{\mathsf{skew}}$. Following [30], we first rewrite the iteration matrix

$$\mathcal{G} \equiv \mathcal{P}^{-1}(\mathcal{P} - \mathcal{A}) \tag{2.1}$$

of the scheme (1.2) as

$$\mathcal{G} = \begin{bmatrix} X(P-A) & 0 \\ Y(P-A) & 0 \end{bmatrix}, \tag{2.2}$$

$$\begin{aligned} \mathbb{R}^{n\times n} \ni X &= P^{-1} - P^{-1}B^T \left(BP^{-1}B^T + C\right)^{-1} BP^{-1}, \\ \mathbb{R}^{m\times n} \ni Y &= \left(BP^{-1}B^T + C\right)^{-1} BP^{-1}. \end{aligned} \tag{2.3}$$

Assume now that P is a positive real matrix. As we will see, for the skew preconditioner (1.5) this will be guaranteed by choosing parameter ω. To get a sufficient condition for convergence, we estimate the spectral radius of $\mathcal{G}$ as

$$\rho(\mathcal{G}) = \rho(X(P-A)) \leqslant \|X(P-A)\|_*. \tag{2.4}$$

It is convenient here to define the norm $\|\cdot\|_*$ as the Euclidean matrix norm with respect to the symmetric part of P:

$$\|X(P-A)\|_* = \left\|P_H^{1/2} X(P-A) P_H^{-1/2}\right\|_2, \qquad P_H \equiv \frac{1}{2}(P+P^T).$$

With this choice of the norm, (2.4) leads to

$$\rho(\mathcal{G}) \leqslant \|X(P-A)\|_* \leqslant \left\|P_H^{1/2} X P_H^{1/2}\right\|_2 \left\|P_H^{-1/2}(P-A)P_H^{-1/2}\right\|_2, \tag{2.5}$$

where $P-A$ and, hence, $P_H^{-1/2}(P-A)P_H^{-1/2}$ are symmetric. This is achieved by choosing P in such a way that its skew-symmetric part is that of A. (Nonsymmetric preconditioners with the property $P - P^T = A - A^T$ were introduced in [39]). The following result holds:

LEMMA 2.1. *Let X be defined by* (2.3)*, with positive semidefinite matrix C, and P_H (the symmetric part of P) be positive definite. Then*

$$\left\|P_H^{1/2} X P_H^{1/2}\right\|_2 \leqslant 1, \tag{2.6}$$

so that (2.5) *leads to*

$$\rho(\mathcal{G}) \leqslant \left\|P_H^{-1/2}(P-A)P_H^{-1/2}\right\|_2. \tag{2.7}$$

Proof. We first consider the case C is positive definite and rewrite matrix X in the form

$$\begin{aligned} X &= P^{-1}\left[P - B^T(BP^{-1}B^T + C)^{-1}B\right]P^{-1} \\ &= P^{-1} - P^{-1}B^TC^{-1}(BP^{-1}B^TC^{-1} + I)^{-1}BP^{-1}. \end{aligned}$$

The Sherman-Morrison-Woodbury formula [29] reduces the last expression to

$$X = (P + B^TC^{-1}B)^{-1},$$

so that, with $P_S \equiv \frac{1}{2}(P - P^T)$ being the skew-symmetric part of P,

$$\begin{aligned} &P_H^{1/2} X P_H^{1/2} = \\ &= \Big(I + \underline{P_H^{-1/2} P_S P_H^{-1/2} + (BP_H^{-1/2})^T C^{-1} (BP_H^{-1/2})}\Big)^{-1} = (I + \underline{M})^{-1}, \end{aligned} \tag{2.8}$$

where the underlined expression is denoted by M. Since C^{-1} is positive definite, the matrix $(BP_H^{-1/2})^T C^{-1}(BP_H^{-1/2})$ is positive semidefinite. Furthermore, $P_H^{-1/2}P_S P_H^{-1/2}$ is skew-symmetric and, therefore, the matrix M is nonnegative real, i.e. $(Mx, x) \geqslant 0$ for any $x \in \mathbb{R}^n$. Hence, $\|(I+M)^{-1}\|_2 \leqslant 1$, and the statement is proven.

In the case C has zero eigenvalues, we write

$$C = U\Lambda U^T,$$

where columns of U are orthonormal eigenvectors of C and $\Lambda = \mathrm{Diag}(\lambda_i^{(c)})$, with $\lambda_i^{(c)}$ being the eigenvalues of C. We introduce, for $\epsilon \geqslant 0$, matrices

$$\Lambda_\epsilon = \Lambda + \epsilon I, \qquad X_\epsilon \equiv P^{-1} - P^{-1}B^T \left(BP^{-1}B^T + U\Lambda_\epsilon U^T\right)^{-1} BP^{-1}.$$

Denoting

$$\tilde{B} \equiv U^T BP_H^{-1/2}, \qquad \tilde{P}_S \equiv P_H^{-1/2}P_S P_H^{-1/2}, \qquad \tilde{\Lambda}_\epsilon^{-1} = \mathrm{Diag}\left(\frac{\epsilon}{\lambda_i^{(c)} + \epsilon}\right) \quad (\epsilon > 0),$$

and using (2.8), we get

$$\begin{aligned}
&P_H^{1/2}X_\epsilon P_H^{1/2} = (I + \tilde{P}_S + \frac{1}{\epsilon}\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B})^{-1} \quad (\epsilon > 0),\\
&\|P_H^{1/2}X_\epsilon P_H^{1/2}\|_2^2 = \max_{\|x\|=1} \frac{1}{((I + \tilde{P}_S + \frac{1}{\epsilon}\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B})x, (I + \tilde{P}_S + \frac{1}{\epsilon}\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B})x)}\\
&\leqslant \frac{1}{1 + 2\min\limits_{\|x\|=1}(\tilde{P}_S x, x) + \frac{1}{\epsilon}\cdot\min\limits_{\|x\|=1}\left[(\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B}x, x) + \frac{1}{\epsilon}\|\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B}x\|^2\right]}\\
&= \frac{1}{1 + 2\min\limits_{\|x\|=1}(\tilde{P}_S x, x) + \frac{1}{\epsilon}\cdot 0} \leqslant 1,
\end{aligned}$$

because the matrix $\tilde{B}^T\tilde{\Lambda}_\epsilon^{-1}\tilde{B}$ has at least $n - m$ zero eigenvalues. We have

$$\|P_H^{1/2}X_\epsilon P_H^{1/2}\|_2^2 \leqslant \frac{1}{1 + 2\min\limits_{\|x\|=1}(\tilde{P}_S x, x)} \leqslant 1.$$

Letting $\epsilon \to 0$ in the last estimate yields

$$\|P_H^{1/2}XP_H^{1/2}\|_2 = \lim_{\epsilon\to 0}\|P_H^{1/2}X_\epsilon P_H^{1/2}\|_2 \leqslant 1.$$

□

We further proceed similarly to [39, 40, 6] and estimate the norm in (2.7). Recall that the matrix $P - A$ is symmetric, so that $P - A = P_H - H$ and

$$\left\|P_H^{-1/2}(P-A)P_H^{-1/2}\right\|_2 = \left\|I - P_H^{-1/2}HP_H^{-1/2}\right\|_2 = \rho(I - P_H^{-1/2}HP_H^{-1/2}), \qquad (2.9)$$

Representing the eigenvalues of $I - P_H^{-1/2}HP_H^{-1/2}$ as Rayleigh quotients, it is easy to see that they are inside the interval $(-1, 1)$ if and only if

$$(0 <) \quad (Hx, x) < 2(P_H x, x) \quad \forall x \in \mathbb{R}^n, \quad x \neq 0. \qquad (2.10)$$

We summarize this section with the following result:

THEOREM 2.2. *Let $\mathcal{G}$ be the iteration matrix of* (1.2), (1.3), *where P is a positive real matrix and $P - P^T = A - A^T$. Then we have that (see Lemma 2.1 and* (2.7)*)*

$$\rho(\mathcal{G}) \leqslant \left\| P_H^{-1/2}(P - A)P_H^{-1/2} \right\|_2,$$

where P_H is the symmetric part of P. Moreover,

$$\left\| P_H^{-1/2}(P - A)P_H^{-1/2} \right\|_2 < 1$$

if and only if inequality (2.10) *holds true.*

2.2. Choice of ω. In this section we adopt, with some minor changes, the results from [39, 40, 6] on how the parameter ω should be chosen. So far we have not used the particular form (1.5) of P. It is easy to check that the symmetric part of P is given by

$$P_H = \frac{1}{\omega}I + \omega L_S U_S = \frac{1}{\omega}I - \omega L_S L_S^T. \tag{2.11}$$

Here and elsewhere in this paper we assume that $\omega > 0$. The following obvious lemma follows:

LEMMA 2.3. *The extremal eigenvalues of the symmetric part P_H of preconditioner* (1.5) *are given by*

$$\begin{aligned} \lambda_{\mathsf{min}}(P_H) &= \min_{\|x\|=1}(P_H x, x) = \frac{1}{\omega} - \omega\|L_S\|_2^2, \\ \lambda_{\mathsf{max}}(P_H) &= \max_{\|x\|=1}(P_H x, x) = \frac{1}{\omega}. \end{aligned} \tag{2.12}$$

Thus, P is positive real, i.e. $\lambda_{\mathsf{min}}(P_H) > 0$, if and only if

$$\omega < \frac{1}{\|L_S\|_2}. \tag{2.13}$$

Taking into account (2.10), we could get conditions on ω (cf. [39, 40, 6]) which would be sufficient for convergence of the iteration (1.2), (1.3), (1.5) provided that (estimates for) the extremum eigenvalues of H and S are known. A similar technique for the classical SSOR, also based on the extremum eigenvalue estimates, has been used in [58, 2] where in particular the norm $\|L_S\|_2$ appears to be an important parameter, too. Indeed, let

$$\lambda_{\mathsf{min}} = \min_{\|x\|_2=1}(Hx, x) = \gamma_1, \qquad \lambda_{\mathsf{max}} = \max_{\|x\|_2=1}(Hx, x) = \gamma_2, \qquad \|L_S\|_2 = \gamma_3.$$

Note that $\rho(S) < 2\gamma_3$. Then, requiring that the maximum of the left-hand side in (2.10) is smaller than the minimum of the right-hand side yields the following condition

$$\gamma_2 < 2(\frac{1}{\omega} - \gamma_3^2\omega), \tag{2.10'}$$

which is sufficient for (2.10) and can easily be solved in ω. Moreover, using the value of γ_1, by a simple field-of-value technique one can minimize an upper bound for the norm in the right-hand side of (2.7) with respect to ω [39, 40, 6]:

$$\begin{aligned}\rho(\mathcal{G}) &\leqslant \|I - P_H^{-1/2} H P_H^{-1/2}\|_2 \\ &= \max\left\{|1-\lambda_{\min}(P_H^{-1/2} H P_H^{-1/2})|, |1-\lambda_{\max}(P_H^{-1/2} H P_H^{-1/2})|\right\} \\ &\leqslant \max\left\{|1-\gamma_1\omega|, \left|1-\frac{\gamma_2\omega}{1-\gamma_3^2\omega^2}\right|\right\} \to \min_{\omega}.\end{aligned} \tag{2.14}$$

It is not difficult to see that the minimum is attained when ω satisfies

$$2-(\gamma_1+\gamma_2)\omega - 2\gamma_3^2\omega^2 + \gamma_1\gamma_3^2\omega^3 = 0,$$

and it is the only real root of this polynomial in the interval $0 < \omega < (\sqrt{\gamma_2^2+16\gamma_3^2}-\gamma_2)/(4\gamma_3^2)$. However, this "optimal" value of ω is typically useless in practice since it essentially optimizes an upper bound of $\rho(\mathcal{G})$ which is not sharp (sharpness is lost in (2.10′) and (2.14)).

2.3. Choice of ω in practice. There are simpler ways to choose ω that usually work well in practice. Let

$$\omega_* = 1/\max\{\|L_S\|_\infty, \|U_S\|_\infty\}. \tag{2.15}$$

It is easy to check that for $\omega < \omega_*$ the symmetric part P_H of the matrix P has diagonal dominance and hence P is positive real. This is necessary for the iteration (1.2), (1.3), (1.5) to converge (Theorem 2.2). Another, slightly sharper computable bound on ω (under which P is positive real) is

$$\omega < \left[\sqrt{\|L_S\|_\infty \, \|U_S\|_\infty}\right]^{-1} = \left[\sqrt{\|L_S\|_\infty \, \|L_S\|_1}\right]^{-1}.$$

Indeed, this last condition implies (2.13) because $\|L_S\|_2 \leqslant \sqrt{\|L_S\|_\infty \, \|L_S\|_1}$.

As numerical experiments suggest, fastest in terms of iteration number convergence is typically observed for values of ω usually ≈ 10 % larger than ω_*. This conclusion is made in [6] for linear systems $Ax = f$ stemming from convection-diffusion problems solved by the Richardson and the GMRES methods preconditioned with P from (1.5). For our problem (1.1) we observe the same dependence of convergence on ω for the GMRES method preconditioned by $\mathcal{P}$ with both the SSOR and the skew blocks P (cf. (1.3), (1.4), (1.5)). The typically observed dependence of the number of iterations (to achieve certain residual reduction) on ω is plotted in Figure 2.1.

Figure 2.2 shows how choice of ω usually influences the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$. Taking ω slightly larger than ω_* typically leads to a condensation of eigenvalues around point $1 + 0i$ in the complex plane and to a separation of several larger eigenvalues on the real axis. Since clustering is generally beneficial for convergence [55, 56], it is natural to expect a faster convergence for this case. Yet further increasing ω results in the eigenvalues with a negative real part and poor convergence.

We emphasize that the preconditioners (1.3),(1.4) and (1.3),(1.5) usually exhibit very robust convergence behavior with respect to perturbations in ω, especially when applied in combination with a modern Krylov subspace method. Taking ω in the neighborhood of ω_* is normally a very good choice in practice. In the numerical

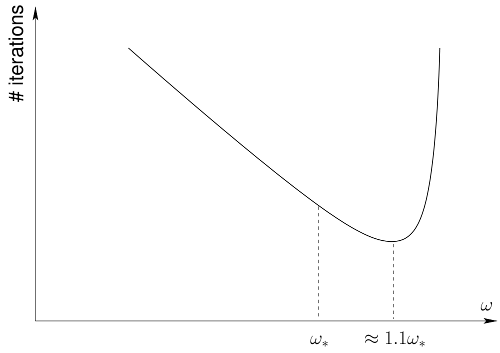

FIG. 2.1. *Typical dependence on ω of the iteration number to achieve a certain residual norm reduction for the skew- and SSOR-preconditioned iterative methods (1.2) observed when solving the test problems described in Section 3. Similar dependence is observed in [6] for central finite difference discretizations of convection-diffusion problems.*

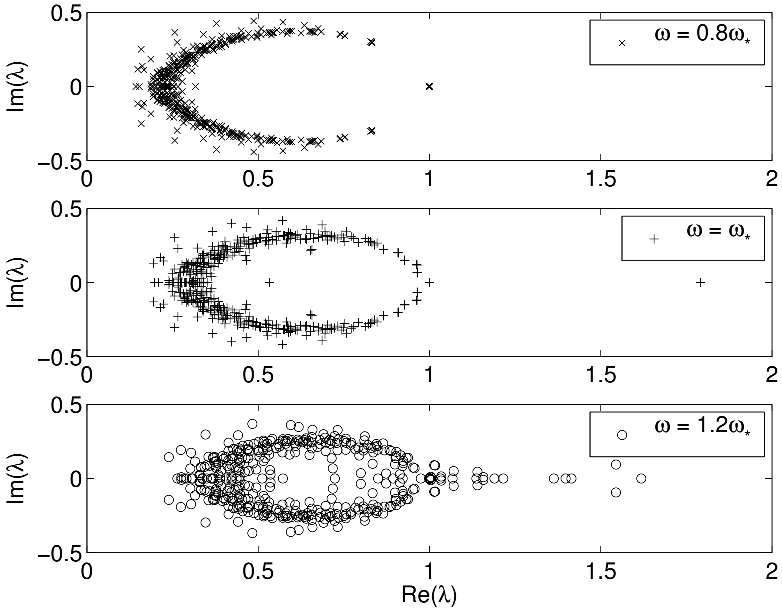

FIG. 2.2. *Eigenvalues of the skew-preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ in the complex plane, for $\omega = 0.8\omega_*$ (top), $\omega = \omega_*$ (middle), $\omega = 1.2\omega_*$ (bottom), ω_* is given by* (2.15). *Eigenvalue clustering and faster convergence are observed for $\omega > \omega_*$. $\mathcal{A}$ is a diagonally scaled discretized Navier-Stokes operator obtained by the stable (Q_1-iso-Q_2)-P_0 discretization (see Section 3) on a* 16×16 *mesh* ($n = 578$, $m = 64$)*, viscosity* $\nu = 0.01$.

experiments presented in Section 3 with the preconditioners (1.3), (1.4) and (1.3), (1.5), ω was chosen as

$$\omega := \begin{cases} [0.9 \max\{\|L_S\|_\infty, \|U_S\|_\infty, 1.0\}]^{-1}, & \text{if } P = P_{\mathsf{skew}}, \\ [0.9 \max\{\|L\|_\infty, \|U\|_\infty, 1.0\}]^{-1}, & \text{if } P = P_{\mathsf{ssor}}, \end{cases} \tag{2.16}$$

where a care is taken that ω does not get too large.

2.4. Preconditioning for a model problem. Here we inspect the preconditioning effect for a simple situation where the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ (cf. (1.3), (1.4)) can be computed analytically. More specifically, we make the following assumption:

Assumption 1. *$\mathcal{A}$ is similar to a block-diagonal matrix with 3×3 diagonal blocks*

$$\begin{bmatrix} 1 & \sigma_i & \hat{b}_{i,1} \\ -\sigma_i & 1 & \hat{b}_{i,2} \\ \hat{b}_{i,1} & \hat{b}_{i,2} & -c_i \end{bmatrix}. \tag{2.17}$$

When applied to matrix $\mathcal{A}$ in the transformed block-diagonal form with blocks (2.17), preconditioning (1.3) results in matrix $\mathcal{P}^{-1}\mathcal{A}$ of the same block-diagonal structure, so that the effect of the preconditioning can be traced for each of the blocks separately. Note that for block-diagonal matrices with blocks (2.17) the SSOR and skew preconditioners coincide (cf. (1.4) and (1.5)).

The following lemma shows that Assumption 1 holds true for a class of matrices $\mathcal{A}$ if B has a special sparsity structure and C is diagonal.

Lemma 2.4. *Let*

$$n = 2m, \quad A = \begin{bmatrix} I & G^T \\ -G & I \end{bmatrix}, \quad R = \begin{bmatrix} V & 0 \\ 0 & U \end{bmatrix}, \quad G, U, V \in \mathbb{R}^{m\times m},$$

where the orthogonal matrices U and V define the singular value decomposition of $G = U\Sigma V^T$. Then there exists a permutation matrix P such that the matrix $(R\mathsf{P})^T AR\mathsf{P}$ is block-diagonal with 2×2 diagonal blocks

$$\begin{bmatrix} 1 & \sigma_i \\ -\sigma_i & 1 \end{bmatrix}, \tag{2.18}$$

with σ_i being the singular values of G. Moreover, Assumption 1 holds true if, in addition, C is diagonal and B is such that the matrix $BR\mathsf{P}$ has nonzero entries only at the positions $(1,1)$, $(1,2)$, $(2,3)$, $(2,4)$, $\ldots$, $(m, 2m-1)$, $(m, 2m)$.

Proof. The proof is straightforward and shows how, under the assumptions of the Lemma, reduction of $\mathcal{A}$ to the block-diagonal form with blocks (2.17) can be made. First, we note that

$$\mathcal{R}^T\mathcal{A}\mathcal{R} = \begin{bmatrix} R^TAR & (BR)^T \\ BR & -C \end{bmatrix} \quad \text{with} \quad \mathcal{R} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}, \quad R^TAR = \begin{bmatrix} I & \Sigma \\ -\Sigma & I \end{bmatrix}.$$

It is not difficult to see that a permutation matrix P exists such that $\mathsf{P}(R^TAR)\mathsf{P} = (R\mathsf{P})^TAR\mathsf{P}$ is block-diagonal with diagonal blocks (2.18). Then

$$(\mathcal{R}\mathbf{P}_1)^T\mathcal{A}\mathcal{R}\mathbf{P}_1 = \begin{bmatrix} (R\mathsf{P})^TAR\mathsf{P} & (BR\mathsf{P})^T \\ BR\mathsf{P} & -C \end{bmatrix}, \qquad \mathbf{P}_1 = \begin{bmatrix} \mathsf{P} & 0 \\ 0 & I \end{bmatrix}.$$

Define now another permutation matrix $\mathbf{P}_2 \in \mathbb{R}^{(n+m)\times(n+m)}$ with columns being those of the identity matrix written in the order

$$1, 2, n+1, 3, 4, n+2, \ldots, n-1, n, n+m.$$

Matrix $(\mathcal{R}\mathbf{P}_1\mathbf{P}_2)^T\mathcal{A}\mathcal{R}\mathbf{P}_1\mathbf{P}_2$ has the required block-diagonal structure with diagonal blocks (2.17). □

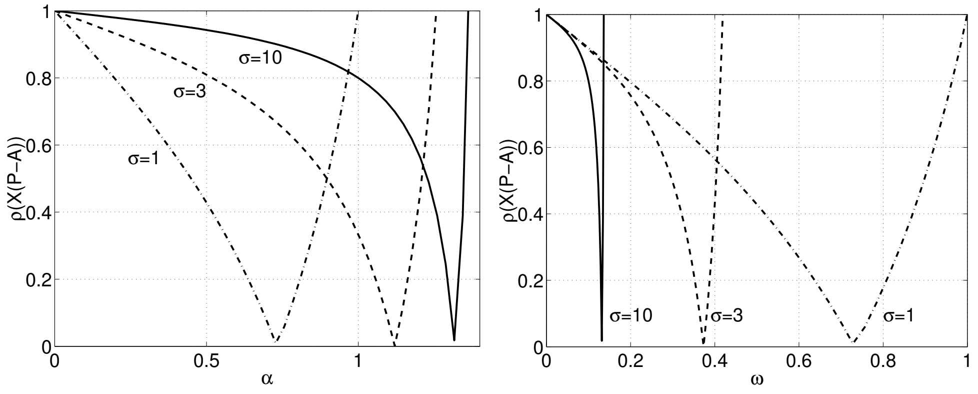

FIG. 2.3. *Model problem, $c = \gamma = 0$. Left plot: Analytically computed $\rho(X(P-A))$ versus $\alpha = \omega\sigma$ for different values of σ. If σ is large enough then the fastest convergence ($\rho(X(P-A)) \approx 0$) takes place for $\alpha > 1$ (for $\omega > \omega_*$). Right plot: the same values against ω. Here $\omega_* \equiv 1/\sigma_{\mathsf{max}} = 0.1$.*

From (2.2), we see that m eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are equal to one and the other n are of the form $1 - \lambda_i$ with λ_i being the eigenvalues of the matrix $X(P-A)$. For the blocks (2.17), computations in Maple show that (we omit the subindices i in (2.17))

$$\begin{aligned} X(P-A) = {} & \frac{1}{\hat{b}_1^2\sigma^2\omega^3 - (\hat{b}_1^2 + \hat{b}_2^2)\omega - c} \\ & \times \begin{bmatrix} (\omega-1)(-\hat{b}_2^2\omega + c(\sigma^2\omega^2 - 1)) & -\omega(\sigma^2\omega^2 - 1 + \omega)(\sigma c + \hat{b}_1\hat{b}_2) \\ \omega(\omega-1)\omega(-\hat{b}_1\hat{b}_2 + \sigma c) & (\sigma^2\omega^2 - 1 + \omega)(\hat{b}_1^2\omega + c) \end{bmatrix}. \end{aligned} \tag{2.19}$$

Without loss of generality we assume that $\sigma > 0$. Since we are mainly interested in the situations for which the skew-symmetric component S is large in norm, we can expect values of σ to be relatively large too, in fact, they are proportional to a norm of S.

For simplicity we consider the case $\hat{b}_1 = \hat{b}_2 = \hat{b}$. It is reasonable to choose σ as a characteristic scale and express ω, $\hat{b}$ and c in terms of σ as

$$\omega = \alpha \left[\frac{1}{\sigma}\right], \qquad \hat{b} = \beta\sigma, \qquad c = \gamma\sigma,$$

where $\alpha = 1$ corresponds to the important choice $\omega = \omega_* \equiv 1/\sigma$ (cf. (2.15)). Note that $\alpha > 0$, $\gamma \geqslant 0$.

In the case $c = \gamma = 0$, the eigenvalues of (2.19) take an elegant form

$$\lambda_1 = 1 + \frac{2\alpha}{\sigma(\alpha^2 - 2)}, \qquad \lambda_2 = 0, \tag{2.20}$$

delivering, for the following two interesting choices of α,

$$\begin{aligned} \lambda_1 &= 1 - \frac{2}{\sigma}, \quad \text{if } \alpha = 1 \ (\omega = \omega_*), \\ \lambda_1 &= 1 + \frac{2}{\sigma^2 - 2}, \quad \text{if } \alpha = \sigma \ (\omega = 1). \end{aligned} \tag{2.21}$$

Note that $\hat{b}$ has no influence on λ_1 at all. Requirement $|\lambda_1| < 1$ is equivalent to

$$0 < \alpha < \bar{\alpha} \equiv \sqrt{2 + \left(\frac{1}{2\sigma}\right)^2} - \frac{1}{2\sigma},$$

where $\bar{\alpha}$ increases monotonically with σ, and $\lim_{\sigma\to 0+}\bar{\alpha} = 0$, $\lim_{\sigma\to\infty}\bar{\alpha} = \sqrt{2}$. Figure 2.3 shows the dependence of $\rho(X(P-A)) = |\lambda_1|$ on α and ω. Here, we recognize the familiar dependence of convergence rate on ω (cf. Figure 2.1). The right plot in Figure 2.3 shows the influence of different blocks (2.17) on the spectral radius: choosing the block with the largest $\sigma = \sigma_{\mathsf{max}}$ and setting $\omega := \omega_* \equiv 1/\sigma_{\mathsf{max}}$ will result for the other blocks in

$$\alpha = \frac{\sigma}{\sigma_{\mathsf{max}}} \quad\Rightarrow\quad \rho(X(P-A)) = |\lambda_1| = \left|1 + \frac{2\sigma/\sigma_{\mathsf{max}}}{\sigma((\sigma/\sigma_{\mathsf{max}})^2 - 2)}\right| < 1. \tag{2.22}$$

For further analysis, where we inspect the effect of the β and γ on the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$, see Appendix.

2.5. Inexact preconditioning. In this section we analyze an inexact form of the preconditioned basic iteration (1.2), (1.3). To compute $v := \mathcal{P}^{-1}u$ (note that we do not compute the matrix $\mathcal{P}^{-1}$ explicitly), an $m \times m$ linear system of the form

$$(BP^{-1}B^T + C)y_2 = y_1, \tag{2.23}$$

has to be solved for a given vector y_1. The inexact method we consider is of interest when solution of (2.23) by a direct method is not feasible and an iterative method is used. This leads to an inner-outer iterative procedure. A proper stopping criterion for the inner iteration should be chosen, for which, on the one hand, the outer iteration convergence is not corrupted too much and, on the other hand, not too many inner iterations are done. The inner-outer iterations have been studied in the context of different problems (see e.g. [28, 26, 22, 3, 57, 52] and references therein). In particular, it is known that if the residual norm tolerance used in the inner iteration converges to zero then usually the convergence rate of the exact method is asymptotically recovered [28, 22, 3]. We will show that this is true for the inexact iteration (1.2).

We first adopt some of the known results on the inner-outer iteration to the outer iteration (1.2) written as

$$u^{k+1} = u^k + \mathcal{P}^{-1}r^k, \qquad r^k \equiv b^k - \mathcal{A}u^k. \tag{2.24}$$

The following simple result provides one possible choice of the inner iteration tolerance for which the convergence rate of the exact method is asymptotically recovered (cf. Theorem 3.3 in [3]):

LEMMA 2.5. *Assume that iteration* (2.24) *converges,* $\rho(\mathcal{G}) < 1$, $\mathcal{G} = \mathcal{P}^{-1}(\mathcal{P}-\mathcal{A})$, *and let* $\|\cdot\|_*$ *be such a norm such that* $\|\mathcal{G}\|_* < 1$. *Assume that at each step of iteration* (2.24) *linear system* $\mathcal{P}(u^{k+1}-u^k) = r^k$ *is solved inexactly, with a residual* $p^k \equiv r^k - \mathcal{P}(u^{k+1}-u^k)$, *so that the inexact iteration reads*

$$u^{k+1} = u^k + \mathcal{P}^{-1}r^k - \mathcal{P}^{-1}p^k. \tag{2.25}$$

Then the inexact iteration (2.24) *converges to the exact solution* $\hat{u}$ *of* (1.1) *in the norm* $\|\cdot\|_*$ *provided that*

$$\begin{aligned} &\|p^k\|_* \leqslant \epsilon_k \|r^k\|_*, \qquad k = 0, 1, \ldots \quad \textit{and} \\ &\varsigma + \epsilon_{\mathsf{max}}\theta < 1, \qquad \epsilon_{\mathsf{max}} = \max_k \epsilon_k, \end{aligned}$$

where $\varsigma \equiv \|\mathcal{G}\|_* < 1$ *and* $\theta = \|\mathcal{P}^{-1}\|_* \cdot \|\mathcal{A}\|_*$. *If, furthermore, the inner iteration tolerance* ϵ_k *satisfies*

$$\epsilon_k \leqslant c\delta^{\tau_k}, \tag{2.26}$$

where $c \geqslant 0$ and $\delta \in (0,1)$ are constants and $\tau_k \geqslant 1$ is a nondecreasing sequence such that $\lim_{k\to\infty} \tau_k = \infty$, then the convergence rate is asymptotically the same as for the exact iteration (2.24)*:*

$$\lim_{k\to\infty} \sup \frac{\|u^{k+1} - \hat{u}\|_*}{\|u^k - \hat{u}\|_*} \leqslant \varsigma. \tag{2.27}$$

Proof. Existence of the norm $\|\cdot\|_*$ follows from the fact that for any $\varepsilon > 0$ there exists at least one matrix norm such that $\|\mathcal{G}\|_* \leqslant \rho(\mathcal{G}) + \varepsilon$ (see e.g. Lemma 5.6.10 in [34]). Subtracting the equality $\hat{u} = \mathcal{G}\hat{u} + \mathcal{P}^{-1}b$ from (2.25) we arrive at

$$u^{k+1} - \hat{u} = \mathcal{G}(u^k - \hat{u}) - \mathcal{P}^{-1}p^k,$$

so that

$$\begin{aligned} \|u^{k+1} - \hat{u}\|_* \leqslant & \|\mathcal{G}\|_* \|u^k - \hat{u}\|_* + \|\mathcal{P}^{-1}\|_* \|p^k\|_* \\ \leqslant & \|\mathcal{G}\|_* \|u^k - \hat{u}\|_* + \|\mathcal{P}^{-1}\|_* \epsilon_k \|r^k\|_*. \end{aligned}$$

Since $\|r^k\|_* = \|\mathcal{A}\hat{u} - \mathcal{A}u^k\|_* \leqslant \|\mathcal{A}\|_* \|u^k - \hat{u}\|_*$ we obtain

$$\|u^{k+1} - \hat{u}\|_* \leqslant (\varsigma + \epsilon_k \theta)\|u^k - \hat{u}\|_*,$$

which shows convergence provided $\varsigma + \epsilon_{\mathsf{max}}\theta < 1$. If ϵ_k satisfies (2.26) then

$$\frac{\|u^{k+1} - \hat{u}\|_*}{\|u^k - \hat{u}\|_*} \leqslant \varsigma + c\delta^{\tau_k}\theta.$$

Letting $k \to \infty$ leads to (2.27). Note that for both the inexact and exact iteration the value of the upper limit in (2.27) depends on the initial guess vector u^0 and belongs to the interval $[\rho(\mathcal{G}), \varsigma]$. If $\mathcal{G}$ is nonsingular then the value of the limit is exactly ς (see [33], Exercise 3.2.12). □

We now consider a specific form of the inexact preconditioner (1.2), (1.3) where the system (2.23) is solved approximately. Direct computations show that

$$\mathcal{P}^{-1} = \begin{bmatrix} P^{-1} - P^{-1}B^TW^{-1}BP^{-1} & P^{-1}B^TW^{-1} \\ W^{-1}BP^{-1} & -W^{-1} \end{bmatrix}, \qquad W \equiv BP^{-1}B^T + C.$$

Let $v = (\mathcal{P} - \mathcal{A})u^k + b$ be partitioned as $v^T = (x^T, y^T)$ with x consisting of the first n components of v. In (1.2), (1.3) we have

$$u^{k+1} = \mathcal{P}^{-1}v = \mathcal{P}^{-1}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^TW^{-1}(BP^{-1}x - y) \\ W^{-1}(BP^{-1}x - y) \end{bmatrix}, \tag{2.28}$$

where to compute the action of W^{-1} the linear system (2.23) with the right-hand side $y_1 = BP^{-1}x - y$ is solved. In our inexact version of (1.2), (1.3) we allow for an approximate solution of this system with residual $q^k \equiv BP^{-1}x - y - Wy_2$ so that

$$y_2 = W^{-1}(BP^{-1}x - y) - W^{-1}q^k \approx W^{-1}(BP^{-1}x - y).$$

In this inexact iteration we essentially work with an approximate preconditioner $\hat{\mathcal{P}}_k \approx \mathcal{P}$:

$$\begin{aligned} u^{k+1} &= \hat{\mathcal{P}}_k^{-1}\left[(\mathcal{P}-\mathcal{A})u^k + b\right] \\ &= \hat{\mathcal{P}}_k^{-1}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^T\left[W^{-1}(BP^{-1}x-y) - W^{-1}q^k\right] \\ W^{-1}(BP^{-1}x-y) - W^{-1}q^k \end{bmatrix} \\ &= \begin{bmatrix} P^{-1}x - P^{-1}B^TW^{-1}(BP^{-1}x-y) \\ W^{-1}(BP^{-1}x-y) \end{bmatrix} + \begin{bmatrix} P^{-1}B^TW^{-1}q^k \\ -W^{-1}q^k \end{bmatrix} \\ &= \mathcal{P}^{-1}v + \mathcal{P}^{-1}\begin{bmatrix} 0 \\ q^k \end{bmatrix}. \end{aligned} \tag{2.29}$$

Substituting $v = (\mathcal{P}-\mathcal{A})u^k + b$ into the last expression, we obtain the following formula for the inexact iteration (1.2), (1.3):

$$u^{k+1} = u^k + \mathcal{P}^{-1}r^k + \mathcal{P}^{-1}\begin{bmatrix} 0 \\ q^k \end{bmatrix}. \tag{2.30}$$

Comparing this last expression with the general inexact form of the Richardson method (2.25), we arrive at

THEOREM 2.6. *Assume that iteration* (1.2), (1.3) *converges,* $\rho(\mathcal{G}) < 1$, $\mathcal{G} = \mathcal{P}^{-1}(\mathcal{P}-\mathcal{A})$, *and let* $\|\cdot\|_*$ *be such a norm that* $\|\mathcal{G}\|_* < 1$. *Then the inexact form* (2.29), (2.30) *of iteration* (1.2), (1.3) *where at step* k *the system* (2.23) *is solved approximately with residual* q^k *converges to the exact solution* $\hat{u}$ *of* (1.1) *in norm* $\|\cdot\|_*$ *provided that*

$$\begin{aligned} &\|q^k\|_* \leqslant \epsilon_k\|r^k\|_*, \qquad k = 0, 1, \ldots \quad \text{and} \\ &\varsigma + \epsilon_{\mathsf{max}}\theta < 1, \qquad \epsilon_{\mathsf{max}} = \max_k \epsilon_k, \end{aligned}$$

where $\varsigma \equiv \|\mathcal{G}\|_*$ *and* $\theta = \|\mathcal{P}^{-1}\|_* \cdot \|\mathcal{A}\|_*$. *If, furthermore, the inner iteration tolerance* ϵ_k *satisfies* (2.26) *then the convergence rate of the inexact iteration* (2.29), (2.30) *is asymptotically the same as for the exact iteration* (1.2), (1.3) *and relation* (2.27) *holds.*

Proof. Note that (2.30) is a particular case of (2.25) with $p^k = -\begin{bmatrix} 0 \\ q^k \end{bmatrix}$ and apply Lemma 2.5. Since $\mathcal{G}$ is singular (see (2.2)), the actual convergence rate depends on the initial guess u^0 and can be smaller than ς ([33], Exercise 3.2.12). □

By choosing tolerance in the inner stopping criterion carefully one could also aim at minimizing the overall computational work in the inner-outer iteration rather than at preserving the outer convergence rate [22, 28]. Adopting these strategies to the inexact SSOR iteration is left for future work. We emphasize that another approach, aiming at maintaining convergence through the outer iteration process, has recently been developed [52, 53, 54]. This approach is valid for a more general situation of inexact matrix-vector products [53] and explains a heuristical inner iteration relaxation strategy reported in [7].

2.6. Implementation and costs. The iterative scheme considered in Section 2.1 is in fact a stationary Richardson method applied to the left-preconditioned system $\mathcal{P}^{-1}\mathcal{A}u = \mathcal{P}^{-1}b$. When applying preconditioner (1.3) in combination with this or any other iterative method, one needs to repeatedly compute a result of the action of the matrix $\mathcal{P}^{-1}$ on a given vector. In this section we explain how this can be done. In

(a) compute $v := \mathcal{P}^{-1}u$

```
for a given $u = \begin{bmatrix} x \\ y \end{bmatrix}$:
(1)   $x_1 := P^{-1}x$,
(2)   $y_1 := Bx_1 - y$,
(3)   $y_1 := (BP^{-1}B^T + C)^{-1}y_1$,
(4)   $x_2 := P^{-1}B^T y_1$,
(5)   $x_1 := x_1 - x_2$,
$v := \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$.
```

(b) compute $w := \mathcal{P}^{-1}\mathcal{A}u$

```
for a given $u = \begin{bmatrix} x \\ y \end{bmatrix}$:
(1)   $x_1 := P^{-1}(P - A)x$,
(2)   $y_1 := Bx_1$,
(3)   $y_1 := (BP^{-1}B^T + C)^{-1}y_1$,
(4)   $x_2 := P^{-1}B^T y_1$,
(5)   $x_1 := x_1 - x_2$,
$w := \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$.
```

FIG. 2.4. *Algorithms for preconditioned matrix-vector products.*

addition, implementation of the matrix-vector multiplication $\mathcal{P}^{-1}\mathcal{A}u$ is considered. As it turns out, it can be organized in such a way that, as compared to computing of $\mathcal{P}^{-1}u$, it requires only little extra work. We therefore emphasize that in most cases one should not separate steps $v := \mathcal{A}u$, $w := \mathcal{P}^{-1}v$ but rather combine them in $w := \mathcal{P}^{-1}\mathcal{A}u$.

Consider first the matrix-vector multiplication $v := \mathcal{P}^{-1}u$. In view of (2.28), for u partitioned as $u^T = (x^T, y^T)$, $x \in \mathbb{R}^n$, we can write

$$u^{k+1} = \mathcal{P}^{-1}v = \mathcal{P}^{-1}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^T W^{-1}(BP^{-1}x - y) \\ W^{-1}(BP^{-1}x - y) \end{bmatrix},$$

with $W \equiv BP^{-1}B^T + C$. This leads to the algorithm shown in Figure 2.4(a).

To work out computation of $w := \mathcal{P}^{-1}\mathcal{A}u$, we use (2.1) and (2.2):

$$\mathcal{P}^{-1}\mathcal{A}u = u - \mathcal{G}u = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} X(P - A)x \\ Y(P - A)x \end{bmatrix}.$$

Substituting X and Y from (2.3), we get

$$\begin{aligned} X(P - A)x &= P^{-1}(P - A)x - P^{-1}B^T \underline{Y(P - A)x} \\ &= \left[I - P^{-1}B^T(BP^{-1}B^T + C)^{-1}B\right] P^{-1}(P - A)x, \end{aligned}$$

so that, computing $X(P - A)x$, we get $Y(P - A)x$ as a by-product. The resulting procedure to compute $v := \mathcal{P}^{-1}\mathcal{A}u$ is outlined in Figure 2.4(b).

Note that in both algorithms from Figure 2.4 the inverse matrices at steps (1), (3), and (4) do not have to be computed, instead, one solves linear systems. Steps (1) and (4) in both algorithms can be done efficiently since P is a product of triangular matrices. The most expensive part is step (3). One possible way here is to use a direct linear solver, computing the LU factorization of the $m \times m$ Schur complement once and then reusing it at every step (3). The costs of the preconditioner for this case are given in Table 2.1. If these costs are not feasible, solution in step (3) can be done iteratively. This leads to the inner-outer iterative procedure analyzed in Section 2.5. We further discuss implementation issues for this method in Section 3.1.

Table 2.1

Estimates for the costs of the algorithms for preconditioned matrix-vector products (Figure 2.4). The notation is: L and U have respectively at most l and u nonzero entries per row, B has no more than r nonzero entries in each row or column. The matrix $(BP^{-1}B^T+C)$ is computed and factorized once. It is assumed that operations $x_1 := P^{-1}Ax$ and $x_1 := P^{-1}x$ have roughly the same costs (as is the case when e.g. the Eisenstat's trick is applied [15]). Terms $O(m)$ and $O(n)$ are omitted.

initialization costs:	
computation of $(BP^{-1}B^T+C)$	$2n(l+u+m)+2m^2r$
LU factorization of $(BP^{-1}B^T+C)$	$\frac{2}{3}m^3$
total:	$\frac{2}{3}m^3+2m^2r+2n(l+u+m)$

costs per matrix-vector product $v := \mathcal{P}^{-1}u$ or $w := \mathcal{P}^{-1}\mathcal{A}u$:	
step (see Figure 2.4)	costs
(1)	$2n(l+u)$
(2),(3)	$2m^2+2m(r-1)$
(4)	$2n(l+u+r)$
(5)	n
total:	$2m^2+2n[2(l+u)+r]+2mr$ (terms $O(m)$ and $O(n)$ are neglected)

Compared with other known preconditioners used for systems (1.1), our preconditioning is not too expensive. For example, one matrix-vector product with the BFB^T preconditioning [16] involves solving a linear system with the matrix A. In addition, solving the eigenvalue problem for BB^T (to rearrange the unknowns) may be necessary. In Table 2.2, the costs of the BFB^T preconditioner are given for the case when the matrix A has bandwidth $\sqrt{n}$, the system with A is solved by a band LU direct solver and the eigenvalue problem for BB^T is solved. Comparing the costs of the SSOR and BFB^T preconditioners (Tables 2.1 and 2.2), we see that application of the preconditioner matrix $v := \mathcal{P}^{-1}u$ for BFB^T is approximately three times more expensive than for SSOR, even if one neglects the costs of the matrix A solve in BFB^T. Indeed, these costs can often be reduced by using an (inner) iterative solver. We emphasize that the costs in BFB^T can be reduced by making it inexact [16], just as can be done for the SSOR preconditioner presented in this paper.

Another, block-triangular preconditioner (see [18] and Section 3.3) requires, for the matrix A having bandwidth $\sqrt{n}$, about $2n^2$ operations at its initialization stage for the band LU factorization of A and about $4n\sqrt{n}+2nr$ operations at each iteration for $w := \mathcal{P}^{-1}v$ (Table 2.3). Thus, this preconditioner is much cheaper than either BFB^T or SSOR when m is large (cf. Tables 2.1, 2.2).

In Section 3.3 we also demonstrate the performance of a simple constraint preconditioner (1.3) with P taken to be the identity matrix, which is the diagonal of A if the diagonal prescaling is applied. The costs of this preconditioner are significantly reduced as compared to general constraint preconditioners (Table 2.1) because of a simpler structure of the matrix $BP^{-1}B^T+C = BB^T+C$. If this matrix is a sparse band matrix with bandwidth $\sqrt{n}$ (as is the case in Section 3.3) the initialization costs are approximately $4mn$ operations and every matrix-vector multiplication requires about $4m\sqrt{n}+2(m+n)r$ operations (cf. Tables 2.1, 2.2).

In the symmetric preconditioners [30], one needs to solve a linear system with a symmetric $n\times n$ matrix at each preconditioned matrix vector multiplication, this can often be done with fast direct solvers.

TABLE 2.2

Estimates for the costs of the exact BFB^T preconditioner matrix-vector products. It is assumed that the matrix A has bandwidth $\sqrt{n}$ as is the case for linearized Navier-Stokes problem of Section 3. The notation is the same as in Table 2.1 and s is the maximal number of nonzero entries per row in C. Terms $O(m)$ and $O(n)$ are omitted.

initialization costs:	
LU band factorization of A	$\approx 2n\sqrt{n}\sqrt{n} = 2n^2$
computation of BB^T	$2m^2(2r-1)$
solving eigenproblem for BB^T	$\frac{8}{3}m^3$
total:	$\frac{8}{3}m^3 + 2m^2(2r-1) + 2n^2$

costs per matrix-vector product:	
operation	costs
$v := \mathcal{A}u$	$2n(l+u+r) + 2m(r+s-1)$
$w := \mathcal{P}^{-1}v$	$6m^2 + 2n(l+u+2r) + 2m(r-1) + 2m - 3 + 4n\sqrt{n}$
total for $w := \mathcal{P}^{-1}\mathcal{A}u$	$6m^2 + 2n[2(l+u)+3r] + 2m(2r+s) + 4n\sqrt{n}$ (terms $O(m)$ and $O(n)$ are neglected)

TABLE 2.3

Estimates for the costs of the exact block-triangular preconditioner matrix-vector products. For notation and assumptions see caption of Table 2.2.

initialization costs:	
LU band factorization of A	$\approx 2n\sqrt{n}\sqrt{n} = 2n^2$

costs per matrix-vector product:	
operation	costs
$v := \mathcal{A}u$	$2n(l+u+r) + 2m(r+s-1)$
$w := \mathcal{P}^{-1}v$	$2nr + 4n\sqrt{n}$
total for $w := \mathcal{P}^{-1}\mathcal{A}u$	$2n(l+u+2r) + 2m(r+s) + 4n\sqrt{n}$ (terms $O(m)$ and $O(n)$ are neglected)

3. Numerical experiments.

We have carried out numerical experiments for systems (1.1) coming from the finite-element discretization of the two-dimensional linearized Navier-Stokes equations (the Oseen equations, see e.g. [21, 19, 20, 17]):

$$\begin{cases} -\nu\Delta\boldsymbol{u} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{u} + \nabla p = \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases}$$

where velocity $\boldsymbol{u}$ is the unknown, $\boldsymbol{v}$ is the known velocity from the previous (Picard) iteration, p is pressure, $\nu > 0$ is viscosity. We have chosen this problem for numerical experiments because it is a widely known and well understood test problem. We emphasize that there are a lot of powerful preconditioners which work very well for the systems steming from the Navier-Stokes equations [21, 35, 17, 20], see also the recent survey [5]. Our aim here is to show that our preconditioning approach based on purely algebraic considerations, though apparently not being the best possible choice for this particular problem, may be competetive with other well known preconditioners.

The test problem is the leaky-lid driven cavity problem, as generated by the MATLAB software of David Sivester and Howard Elman [50], with the wind field

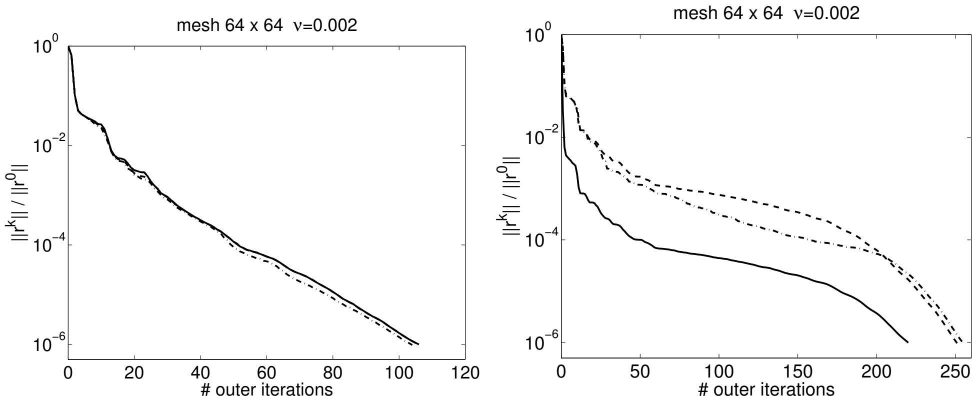

FIG. 3.1. *Convergence plots for preconditioned GMRES with three different Schur complement solvers: direct solver with LU factorization (solid line), inner iteration with the strict stopping criterion $\|q^k\|_2 \leqslant 10^{-6}\|q^0\|_2$ (dashed line), and inner iteration with stopping criterion* (3.1) *(dash-dotted line). Left plot: the stable (Q_1-iso-Q_2)-P_0 discretization. Right plot: the stabilized Q_1-P_0 discretization. In both cases $\nu = 0.002$ and 64×64 mesh is used. For the stable discretization, accuracy of the inner solver has almost no influence on the outer iteration convergence.*

$\boldsymbol{v} = (v_1, v_2)$ chosen as

$$v_1(x, y) = 2y(1 - x^2), \qquad v_2(y) = -2x(1 - y^2), \qquad -1 \leqslant x, y \leqslant 1.$$

The software can produce two types of discretizations: the stable (Q_1-iso-Q_2)-P_0 discretization and the stabilized Q_1-P_0 discretization (in the former case $C = 0$) (see e.g. [19, 51, 35] and references therein). The stabilization parameter for the stabilized Q_1-P_0 discretization was $\beta = 0.25$.

Throughout this section, ω in preconditioners (1.3),(1.4) and (1.3),(1.5) was chosen according to (2.16). For the SSOR, skew and Stokes preconditioners, the two-sided diagonal prescaling was used to get $\mathrm{Diag}(A) = I$. We used full GMRES [47] as the (outer) iterative solver. All the runs were done on a PC with a 2.5 GHz processor and 2 Gb memory.

3.1. Implementation of the inexact iteration. Analysis of Section 2.5 gives convergence conditions for the Richardson iteration (1.2) when action of the Schur complement inverse is computed approximately by another, inner iterative process. Since the obtained conditions (Theorem 2.6) are based on the norm estimates that are not sharp, we expect them to be too strict in practice. These conditions should also be relaxed when a modern Krylov subspace method, instead of the simplest Richardson method, is employed in the outer iteration. Therefore, we interpret Theorem 2.6 qualitatively rather than quantitatively: the residual norm in the inner iteration should be proportional to the outer iteration residual norm.

In our numerical experiments, we have used the following stopping criterion for the inner iterations:

$$\|q^k\|_2 \leqslant \begin{cases} 10^{-6}, & \text{if } \|r^k\|_2 > 0.01, \\ \|r^k\|_2, & \text{otherwise,} \end{cases} \tag{3.1}$$

q^k and r^k are inner and outer residuals, respectively. Here, the strict tolerance on $\|q^k\|_2$ for large outer residual norm is not caused by the convergence requirements but

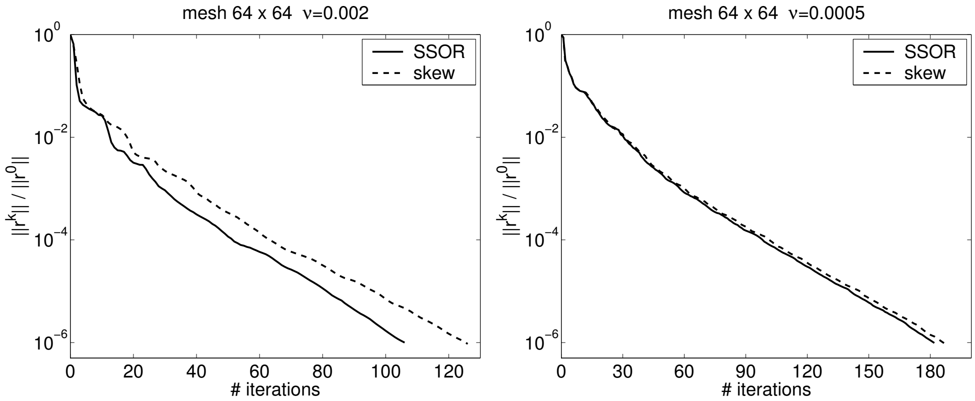

FIG. 3.2. *Convergence plots for GMRES preconditioned with the skew and SSOR preconditioners. Left plot:* 64×64 *mesh,* $\nu = 0.002$*, right plot:* 64×64 *mesh,* $\nu = 0.0005$*. The stable (*Q_1*-iso-*Q_2*)-*P_0 *discretization is used.*

rather by accuracy requirements of this specific test problem. Without this condition, convergence of the outer process is not corrupted. However, the obtained solution is much less accurate, as compared to the known exact solution. We do not have an exact explanation for this. For the inner iterative solver full GMRES was taken with the incomplete LU factorization of the matrix $BB^T + C$ as the preconditioner. Note that in case $BB^T + C$ is guaranteed to be positive definite the incomplete Cholesky factorization could be used as a preconditioner. The ILU preconditioner is useful but by no means crucial for the overall performance. The maximum number of iterations was taken 15 and 50 respectively for the stable (Q_1-iso-Q_2)-P_0 and for the stabilized Q_1-P_0 discretizations. A lower value of the maximum iteration number for the stable discretization was taken for efficiency reasons, because of a very robust outer iteration convergence behavior observed in this case.

An example showing how the chosen inner stopping criterion affects the outer iteration convergence can be seen in Figure 3.1, where, in addition to the stopping criterion (3.1), convergence plots for a much stricter criterion are given. As we see, for the stable discretization the outer iteration convergence is hardly affected by the choice of the inner solver. This robust convergence behavior was observed for this discretization in almost all runs (see the results reported in Section 3.3).

3.2. Skew and SSOR preconditioners. The convergence results of Section 2.1 were obtained for the skew preconditioner (1.3), (1.5) rather than for the SSOR preconditioner (1.3), (1.4) we are aiming at. As it has been already stated, the skew preconditioner should be an increasingly better approximation to SSOR as the skew-symmetric component S of A grows in norm. In practice, we do observe the similar convergence behavior of both preconditioners already for not so small viscosity values ν (note that $\|S\| \sim 1/\nu$). This can be seen in Figure 3.2 and in the results reported in Section 3.3.

3.3. Comparison with other preconditioners. Here, we present results of the comparison of the SSOR and skew preconditioners with the BFB^T preconditioner [16], the block-triangular preconditioner [18], and the Stokes preconditioner [30]. Together with the "exact" version of the SSOR preconditioner, we test the inexact SSOR preconditioner implemented as explained in Section 3.1.

We emphasize that comparison of inexact forms of SSOR with inexact forms of other preconditioners is beyond the scope of the paper. We restrict ourselves to demonstrating that
(i) the new preconditioning technique in its exact form can be very competitive with other techniques for the cases where the exact implementations are feasible and require comparable work,
(ii) the inexact form of the new preconditioner (presented in Section 3.1) is robust and leads to only a moderate increase of the number of the outer iterations.

In the Stokes preconditioner, $\mathcal{P}$ is taken in the same way as in (1.3), with P being the symmetric component H of A. Therefore, we expect this preconditioner to work well only for weakly nonsymmetric systems (for large viscosity values). We implement the Stokes preconditioner by computing the Cholesky factorization of H and then following the same procedure as for the SSOR and skew preconditioners (see Section 2.6).

The BFB^T and the block-triangular preconditioners were used for respectively the stable (Q$_1$-iso-Q$_2$)-P$_0$ and for the stabilized Q$_1$-P$_0$ discretizations. These preconditioners were selected for comparison as the most efficient preconditioners provided by the software [50] for each of these discretizations.

The costs of the SSOR, BFB^T and block-triangular preconditioners, as they are implemented for this test, are reported in Tables 2.1–2.3. For this model problem the parameters appearing in the cost estimations have the following values: $l = u = 4$, $r = 10$, $s = 0$ for the stable discretization, and $s = 3$ for the stabilized discretization. For the values of m and n see Tables 3.1 and 3.2.

Both the BFB^T and the block-triangular preconditioners involve action of A^{-1} on a given vector, this was implemented by the LU factorization (which was computed once and reused every iteration, see Tables 2.2 and 2.3). This factorization and corresponding back/forward substitutions (with costs of $2n^2$ and $4n\sqrt{n}$ operations, respectively) were done very fast, taking a hardly noticeable part of the total CPU time.

The SSOR and BFB^T preconditioners both required $O(m^3)$ floating point operations at the initialization stage and an iteration of BFB^T was approximately a factor three more expensive than an SSOR iteration. The initialization stages for both preconditioners took up a significant, sometimes dominant part of the total CPU time. For this reason the reported CPU time is not proportional to the number of iterations.

We have also made tests with a simple constraint preconditioner (1.3) with P taken to be the identity matrix which, due to the applied diagonal prescaling, is the diagonal of A. The matrix $BP^{-1}B^T + C = BB^T + C$ is a sparse band matrix with bandwidth $\sqrt{n}$ and the costs of this preconditioner scale linearly with m (see Section 2.6).

The results of the comparisons are presented in Tables 3.1 and 3.2. As we see in Table 3.1 the SSOR preconditioner competes quite well with the other techniques, especially for the small values of ν, when the matrix A is far from symmetric. For this stable discretization the inexact form is much slower than the exact form (even though the maximum number of the inner iterations was restricted to 15, see Section 3.1). This is because of the high efficiency of direct solvers in MATLAB, two-dimensionality of the test problem and its size. The situation is different for the stabilized discretization (see Table 3.2): in the SSOR, skew and Stokes preconditioners the costs for the Schur complement system solution are increased due to the larger values of m and the simple constraint preconditioner with $P = I$, appears to be the most efficient in

Table 3.1

The CPU time (seconds) and number of iterations (given in brackets) for different preconditioners, mesh sizes and viscosity parameters ν. The stable (Q_1-iso-Q_2)-P_0 discretization. "—" means that a preconditioner has not been tried for this case.
One BFB^T iteration is approximately a factor three more expensive than one SSOR or skew iteration (for estimation of costs per iteration see Tables 2.1 and 2.2). We emphasize that the reported CPU times are obtained for MATLAB codes and, thus, give only an indication of the performance.

$n = 2178$, $m = 256$ (mesh 32×32)

ν	SSOR	inexSSOR	skew	BFB^T	Stokes	$P = I$
0.1	1.4(24)	3.9(24)	2.4(64)	1.6(29)	2.1(16)	2.4(74)
0.01	1.9(41)	4.7(41)	2.3(60)	1.7(32)	3.3(85)	6.8(198)
0.005	2.0(43)	8.5(43)	2.1(50)	2.0(37)	8.7(141)	6.6(198)
0.002	2.6(69)	12(70)	2.7(71)	9.5(111)	16(261)	9.8(277)

$n = 8450$, $m = 1024$ (mesh 64×64)

ν	SSOR	inexSSOR	skew	BFB^T	Stokes	$P = I$
0.1	12(50)	27(50)	18(130)	21(42)	25(15)	13(143)
0.01	14(82)	51(83)	23(150)	22(46)	37(91)	51(437)
0.005	17(116)	68(117)	23(150)	22(45)	53(176)	70(546)
0.002	16(106)	61(104)	20(126)	23(50)	75(277)	54(494)
0.001	20(129)	73(126)	21(137)	46(124)	130(467)	65(568)
0.0005	26(182)	96(182)	27(187)	127(344)	209(693)	101(771)

Table 3.2

The CPU time (seconds) and number of iterations (given in brackets) for different preconditioners, mesh sizes and viscosity parameters ν. The stabilized Q_1-P_0 discretization. "—" means that a preconditioner has not been tried for this case.
For estimation of costs per iteration see Tables 2.1 and 2.3. We emphasize that the reported CPU times are obtained for MATLAB codes and, thus, give only an indication of the performance.

$n = 2178$, $m = 1024$ (mesh 32×32)

ν	SSOR	inexSSOR	skew	block-tr	Stokes	$P = I$
0.1	7.3(23)	5.2(23)	10(51)	0.7(28)	9.0(15)	1.8(76)
0.01	10(55)	19(55)	12(70)	12(286)	21(89)	6.2(231)
0.005	13(84)	28(85)	14(96)	35(549)	29(145)	8.2(326)
0.002	20(154)	50(163)	21(161)	78(796)	35(246)	16(486)

$n = 8450$, $m = 4096$ (mesh 64×64)

ν	SSOR	inexSSOR	skew	block-tr	Stokes	$P = I$
0.1	272(48)	70(48)	470(120)	3.3(25)	280(15)	17(128)
0.01	296(69)	176(90)	414(162)	58(283)	384(90)	81(443)
0.005	509(139)	276(144)	622(159)	193(615)	708(159)	127(665)
0.002	684(220)	476(255)	561(276)	>1075(>1500)	654(270)	285(1002)
0.001	706(371)	705(386)	749(400)	>2500(>2000)	1264(425)	433(1348)
0.0005	963(574)	1097(589)	995(592)	—	—	718(1822)

this case. The performance of this preconditioner, however, deteriorates strongly as ν decreases and A becomes more nonsymmetric. We also see that the inexact SSOR preconditioner is more competitive for this problem.

4. Conclusions and an outlook to future research.

As our analysis and experiments suggest, the nonsymmetric preconditioner approach (1.3), (1.4) appears to be an interesting alternative to other preconditioning techniques, especially when A

is strongly nonsymmetric and when the Schur complement system can be efficiently solved (for example, because of its size or structure). When solution of the Schur complement system is expensive, inexact forms of the preconditioner can be employed. Our experiments show that the chosen simple strategy for the inner-outer iteration (namely, keeping the inner residual norm bounded by the outer iteration residual norm) usually works well in practice. However, if necessary, more can be done to minimize the overall work in the inner-outer iteration (see e.g. [22]).

The framework introduced in Section 2.1 can be applied to analyze any preconditioner (1.3) with P having the same skew-symmetric part as A. In fact, other choices of P are possible. For example, one could define a class of *skew incomplete LU factorizations* of A (cf. (1.4))

$$P := P_{\mathsf{skewILU}} \equiv (D + L_S)D^{-1}(D + U_S),$$

where D is a diagonal matrix chosen such that $\mathrm{Diag}(A) = \mathrm{Diag}(P)$ or, for the modified version of ILU, such that the row sums in A and P are identical. Another class of skew preconditioners for the discretized Navier-Stokes problems can be obtained by using the rotation form of the equations [44, 45, 41]. These skew preconditioners will be a subject of future research.

Acknowledgements. The first author thanks Henk van der Vorst and Gerard Sleijpen for stimulating discussions at early stages of this work. We thank the anonymous referees for helping us to improve this paper and Michele Benzi for useful comments and, in particular, for bringing our attention to [44, 45, 41].

REFERENCES

[1] M. Arioli and L. Baldini. A backward error analysis of a null space algorithm in sparse quadratic programming. *SIAM J. Matrix Anal. Appl.*, 23(2):425–442, 2001.

[2] O. Axelsson. A generalized SSOR method. *Nordisk Tidskr. Informationsbehandling (BIT)*, 12:443–467, 1972.

[3] Z.-Z. Bai, G. H. Golub, and M. K. Ng. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, 24(3):603–626, 2003.

[4] M. Benzi and G. H. Golub. A preconditioner for generalized saddle point problems. *SIAM J. Matrix Anal. Appl.*, 26(1):20–41, 2004.

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

[6] M. A. Bochev and L. A. Krukier. Iterative solution of strongly nonsymmetric systems of linear algebraic equations. *Russian Comput. Mathematics and Math. Physics*, 37(11):1241–1251, 1997.

[7] A. Bouras and V. Frayssé. A relaxation strategy for inexact matrix-vector products for Krylov methods. Report TR/PA/00/15, CERFACS, France, 2000.

[8] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM J. Numer. Anal.*, 34(3):1072–1092, 1997.

[9] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev. Uzawa type algorithms for nonsymmetric saddle point problems. *Math. Comp.*, 69(230):667–689, 2000.

[10] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*. Springer, 2002.

[11] E. de Sturler and J. Liesen. Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. part i: Theory. *SIAM Journal on Sci. Comp.*, 26(5):1598–1619, 2005.

[12] H. S. Dollar and A. J. Wathen. Approximate factorization constraint preconditioners for saddle-point matrices. *SIAM J. Sci. Comput.*, 2005. To appear.

[13] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner. The factorization of sparse symmetric indefinite matrices. *IMA J. Numer. Anal.*, 11(2):181–204, 1991.

[14] N. Dyn and W. E. Ferguson, Jr. The numerical solution of equality constrained quadratic programming problems. *Math. Comp.*, 41(163):165–170, 1983.

[15] S. C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J. Sci. Stat. Comp.*, 2(1):1–4, 1981.

[16] H. C. Elman. Preconditioning for the steady-state Navier-Stokes equations with low viscosity. *SIAM J. Sci. Comput.*, 20(4):1299–1316 (electronic), 1999.

[17] H. C. Elman. Preconditioners for saddle point problems arising in computational fluid dynamics. *Appl. Numer. Math.*, 43, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[18] H. C. Elman and D. J. Silvester. Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations. *SIAM J. Sci. Comput.*, 17(1):33–46, 1996. Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).

[19] H. C. Elman, D. J. Silvester, and A. J. Wathen. Iterative methods for problems in computational fluid dynamics. In *Iterative methods in scientific computing (Hong Kong, 1995)*, pages 271–327. Springer, Singapore, 1997.

[20] H. C. Elman, D. J. Silvester, and A. J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numer. Math.*, 90(4):665–688, 2002.

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

[22] E. Giladi, G. H. Golub, and J. B. Keller. Inner and outer iterations for the Chebyshev algorithm. *SIAM J. Numer. Anal.*, 35(1):300–319, 1998.

[23] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM J. Matrix Anal. Appl.*, 13(1):292–311, 1992.

[24] V. Girault and P.-A. Raviart. *Finite element approximation of the Navier-Stokes equations.* Springer-Verlag, 1979.

[25] R. Glowinski. *Numerical methods for nonlinear variational problems.* Springer Series in Computational Physics. Springer-Verlag, New York, 1984.

[26] G. H. Golub and H. C. Elman. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31(6), 1994.

[27] G. H. Golub and C. Greif. On solving block-structured indefinite linear systems. *SIAM J. Sci.*

Comput., 24(6):2076–2092, 2003.

[28] G. H. Golub and M. L. Overton. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. *Numer. Math.*, 53:571–593, 1988.

[29] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore and London, third edition, 1996.

[30] 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(2):530–539, 1998.

[31] G. H. Golub, X. Wu, and J.-Y. Yuan. SOR-like methods for augmented systems. *BIT*, 41(1):71–85, 2001.

[32] 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(4):1376–1395, 2001.

[33] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, 1994.

[34] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1986.

[35] D. Kay, D. Loghin, and A. J. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[36] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1300–1317, 2000.

[37] A. Klawonn. An optimal preconditioner for a class of saddle point problems with a penalty term. *SIAM J. Sci. Comput.*, 19(2):540–552, 1998.

[38] A. Klawonn and G. Starke. Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis. *Numer. Math.*, 81(4):577–594, 1999.

[39] L. A. Krukier. Implicit difference schemes and an iterative method for solving them for a certain class of systems of quasi-linear equations. *Sov. Math.*, 23(7):43–55, 1979. Translation from Izv. Vyssh. Uchebn. Zaved., Mat. 1979, No. 7(206), 41–52 (1979).

[40] L. A. Krukier. Convergence accelaration of triangular iterative methods based on the skew symmetric part of the matrix. *Appl. Numer. Math.*, 30:281–290, 1999.

[41] G. Lube and M. A. Olshanskii. Stable finite-element calculation of incompressible flows using the rotation form of convection. *IMA J. Numer. Anal.*, 22(3):437–461, 2002.

[42] P. Monk. *Finite Element Methods for Maxwell's Equations.* Oxford University Press, 2003.

[43] J. Nocedal and S. Wright. *Numerical optimization.* Springer-Verlag, 1999.

[44] M. A. Olshanskii. An iterative solver for the Oseen problem and numerical solution of incompressible Navier-Stokes equations. *Numer. Linear Algebra Appl.*, 6(5):353–378, 1999.

[45] M. A. Olshanskii and A. Reusken. Navier-Stokes equations in rotation form: a robust multigrid solver for the velocity problem. *SIAM J. Sci. Comput.*, 23(5):1683–1706, 2002.

[46] W.-Q. Ren and J.-X. Zhao. Iterative methods with preconditioners for indefinite systems. *J. Comput. Math.*, 17(1):89–96, 1999.

[47] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[48] V. Sarin and A. Sameh. An efficient iterative method for the generalized Stokes problem. *SIAM J. Sci. Comput.*, 19(1):206–226, 1998. Special issue on iterative methods (Copper Mountain, CO, 1996).

[49] C. Siefert and E. de Sturler. Preconditioners for generalized saddle-point problems. Report No. UIUCDCS-R-2004-2448, June 2004.

[50] D. J. Silvester and H. C. Elman. Software for the workshop "Theoretical and practical aspects of incompressible CFD". Utrecht University, Mathematical institute, 1997.

[51] D. J. Silvester, H. C. Elman, D. Kay, and A. J. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comput. Appl. Math.*, 128(1-2):261–279, 2001. Numerical analysis 2000, Vol. VII, Partial differential equations.

[52] V. Simoncini and D. B. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numer. Anal.*, 40(6):2219–2239, 2002.

[53] J. van den Eshof and G. L. G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM J. Matrix Anal. Appl.*, 26(1):125–153, 2004.

[54] J. van den Eshof, G. L. G. Sleijpen, and M. B. van Gijzen. Iterative linear solvers with approximate matrix-vector products. In A. Boriçi, A. Frommer, B. Joó, A. D. Kennedy, and B. Pendleton, editors, *QCD and Numerical Analysis III*, volume 47 of *Lecture Notes in Computational Science and Engineering*, pages 133–141, Berlin, Heidelberg, New York, 2005. Springer. Proceedings of the "Third International Workshop on Numerical Analysis and Lattice QCD", Edinburgh, June/July 2003.

[55] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.

[56] H. A. van der Vorst and C. Vuik. The superlinear convergence of GMRES. *J. Comput. Appl.*

Math., 48:327–341, 1993.

[57] H. A. van der Vorst and C. Vuik. GMRESR: a family of nested GMRES methods. *Numer. Lin. Alg. Appl.*, 1:369–386, 1994.

[58] D. M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, 1971.

[59] W. Zulenher. Analysis of iterative methods for saddle point problems: a unified approach. *Math. Comp.*, 71:479–505, 2002.

Appendix. Here, we further analyze the model problem of Section 2.4. For the realistic choice $\omega = \omega_* \equiv 1/\sigma$, we inspect the effect of the β and γ on the eigenvalues of (2.19). This awkward expression reduces in this case to

$$X(P-A) = \begin{bmatrix} \frac{\sigma-1}{\sigma(\gamma+\beta^2)}\beta^2 & \frac{1}{\sigma} \\ \frac{\sigma-1}{\sigma(\gamma+\beta^2)}(\gamma-\beta^2) & -\frac{1}{\sigma} \end{bmatrix}, \tag{2.19$'$}$$

whose eigenvalues are (cf. Figure 4.1):

$$\lambda_{1,2} = \frac{1}{2}\left[\frac{\sigma-1}{\sigma(\gamma+\beta^2)}\beta^2 - \frac{1}{\sigma} \pm \sqrt{\left(\frac{\sigma-1}{\sigma(\gamma+\beta^2)}\beta^2 - \frac{1}{\sigma}\right)^2 + 4\frac{\sigma-1}{\sigma^2(\gamma+\beta^2)}\gamma}\,\right]. \tag{4.1}$$

Analysis of the eigenvalues yields this lemma:

LEMMA 4.1. *Let the matrix $X(P-A))$ given by (2.19′) result from the action of the SSOR preconditioner* (1.3), (1.4) *on the blocks* (2.17), *$\sigma > 1$ and $\omega = \omega_* \equiv 1/\sigma$. Then for the spectral radius of $X(P-A))$ holds:*

1. $\rho(X(P-A)) = |1 - 2/\sigma|$ if $\gamma = 0$.

2. For $\gamma > 0$,

$$\rho(X(P-A)) = \begin{cases} |\lambda_2|, & \text{if} \quad 0 \leqslant |\beta| < \bar\beta, \\ |\lambda_1|, & \text{otherwise}, \end{cases}$$
$$\bar\beta = \begin{cases} \sqrt{\dfrac{\gamma}{\sigma-2}}, & \text{if} \quad \sigma > 2, \\ \infty, & \text{otherwise}. \end{cases} \tag{4.2}$$

Furthermore, $\rho(X(P-A))$ decreases monotonically with $|\beta|$ whenever $\rho(X(P-A)) = |\lambda_2|$ or $\sigma \leqslant 3$. If $\rho(X(P-A)) = |\lambda_1|$ then it is a constant (monotonically increasing) function in $|\beta|$, for $\sigma = 3$ (respectively, for $\sigma > 3$). Finally,

$$\rho(X(P-A)) \leqslant \max\left\{\frac{1+\sqrt{4\sigma-3}}{2\sigma}, 1 - \frac{2}{\sigma}\right\} \quad \text{for any} \quad \beta,\ \gamma > 0,\ \sigma > 1. \tag{4.3}$$

Proof. For $\gamma = 0$, it follows directly from (2.20) that $\rho(X(P-A)) = |1 - 2/\sigma|$. For $\gamma > 0$ we analyze the eigenvalues (4.1) as functions of $|\beta|$. Since β appears in $\lambda_{1,2}$ only as β^2, assume, without loss of generality, that $\beta > 0$. From (4.1), we have

$$\lambda_{1,2} = \frac{1}{2}(f(\beta) \pm \sqrt{f^2(\beta) + g(\beta)}),$$
$$f(\beta) = \frac{\sigma-1}{\sigma(\gamma+\beta^2)}\beta^2 - \frac{1}{\sigma}, \qquad g(\beta) = 4\frac{\sigma-1}{\sigma^2(\gamma+\beta^2)}\gamma.$$

If $\sigma > 1$ then $f'(\beta) > 0$ and derivatives of $\lambda_{1,2}$ with respect to β are positive if and only if

$$\sqrt{f^2(\beta)+g(\beta)} \pm \left(f(\beta) + \frac{g'(\beta)}{2f'(\beta)}\right) = \sqrt{f^2(\beta)+g(\beta)} \pm \left(f(\beta) + \frac{-2}{\sigma}\right) > 0. \tag{4.4}$$

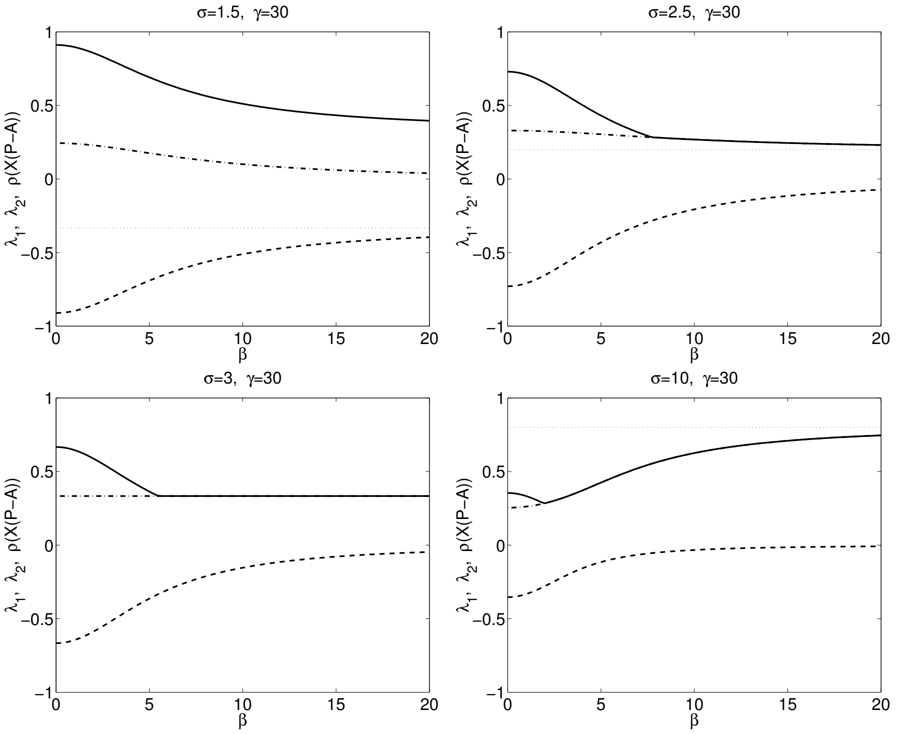

FIG. 4.1. *An illustration of Lemma 4.1: the eigenvalues* λ_1 *(dash-dotted line),* λ_2 *(dashed line) and the spectral radius (solid line) of the matrix* $X(P-A)$ *against* β *for different values of* σ*. The dotted line is the asymptote* $1-\frac{2}{\sigma}$*. The value of* γ *is taken arbitrarily.*

The second of these inequalities (with the minus sign) corresponds to λ_2 and is always true. Hence, λ_2 monotonically increases with β. Multiplying the two inequalities (4.4) with each other, we obtain

$$g(\beta)-\frac{4}{\sigma^2}+\frac{4}{\sigma}f(\beta)>0 \quad \Leftrightarrow \quad (\sigma-3)\gamma>-(\sigma-3)\beta^2,$$

which holds if and only if $\sigma>3$. If $\sigma=3$ then $\lambda_1\equiv 1/3$. Furthermore, it is easy to see that $\lambda_{1,2}$ have different signs and that for $\beta=0$ it holds $-\lambda_2=\frac{1+\sqrt{4\sigma-3}}{2\sigma}>\lambda_1$. Moreover,

$$\lim_{\beta\to\infty}\lambda_1=\max\left\{0,1-\frac{2}{\sigma}\right\}, \qquad \lim_{\beta\to\infty}\lambda_2=\min\left\{0,1-\frac{2}{\sigma}\right\}.$$

This completes the proof. □

The eigenvalues and the spectral radius of $X(P-A)$ from (2.19′) for the different situations described in Lemma 4.1 are plotted in Figure 4.1.

Lemma 4.1 provides information on the action of the SSOR preconditioner (1.3), (1.4) for the choice $\omega:=\omega_*\equiv 1/\sigma$ which is made only for the blocks (2.17) with $\sigma=\sigma_{\mathsf{max}}$. For the other blocks this choice of ω will result in $\omega=\alpha/\sigma$ with $\alpha=\sigma/\sigma_{\mathsf{max}}$. Then the eigenvalues of these blocks are either given by (2.22) for $\gamma=0$ or, for $\gamma>0$, can be obtained in the same way as done in (2.19′), (4.1). The precise analysis of these eigenvalues is rather complicated and beyond the scope of this paper.