SIAM J. MATRIX ANAL. APPL.
Vol. 31, No. 2, pp. 584–601

ITERATIVE SOLUTION OF SKEW-SYMMETRIC LINEAR SYSTEMS*

CHEN GREIF† AND JAMES M. VARAH†

Abstract. We offer a systematic study of Krylov subspace methods for solving skew-symmetric linear systems. For the method of conjugate gradients we derive a backward stable block decomposition of skew-symmetric tridiagonal matrices and set search directions that satisfy a special relationship, which we call skew-A-conjugacy. Imposing Galerkin conditions, the resulting scheme is equivalent to the CGNE algorithm, but the derivation does not rely on the normal equations. We also discuss minimum residual algorithms, review recent related work, and show how the iterations are derived. The important question of preconditioning is then addressed. The preconditioned iterations we develop are based on preserving the skew-symmetry, and we introduce an incomplete 2×2 block LDL^T decomposition. A numerical example illustrates the convergence properties of the algorithms and the effectiveness of the preconditioning approach.

Key words. skew-symmetric, iterative solvers, conjugate gradients, minimum residuals, preconditioners

AMS subject classifications. 65F10, 65F50, 15A57

DOI. 10.1137/080732390

1. Introduction.

A real matrix A is skew-symmetric if $A = -A^T$. Such matrices have a zero main diagonal; all their eigenvalues are pure imaginary; they are necessarily singular if their dimension is an odd number; they are normal and hence unitarily diagonalizable; and their inverse is also skew-symmetric. A brief overview of skew-symmetric matrices and their properties can be found, for example, in [11].

Any real matrix can be split into a symmetric and a skew-symmetric part:

$$A = \left(\frac{A + A^T}{2}\right) + \left(\frac{A - A^T}{2}\right).$$

When the skew-symmetric part, $(A - A^T)/2$, is dominant, it is typically harder to solve the linear system $Ax = b$ or the eigenvalue problem $Ax = \lambda x$. Specifically, the case of A purely skew-symmetric is challenging from a numerical point of view.

For direct solvers of skew-symmetric linear systems, a 2×2 block LDL^T decomposition with symmetric pivoting has been proposed by Bunch [3]. See also [2, 7, 11] for further discussion of direct solution methods.

For iterative solvers, given the symmetric distribution of the eigenvalues over the imaginary axis on both sides of the origin, and the strong dependence of performance of Krylov subspace solution methods on the spectral structure of the matrix, solvers for skew-symmetric systems are expected to perform similarly to solvers for symmetric indefinite systems. By a well-known result of Faber and Manteuffel [8], a short recurrence conjugate gradient (CG) method exists for real skew-symmetric matrices, and we can expect to find a CG-type scheme that minimizes the error in some inner product norm. (See also the discussion in [9, Chap. 6].) However, the concept of an A-norm cannot be directly utilized, as $x^T Ax = 0$ for all real vectors x.

*Received by the editors August 8, 2008; accepted for publication (in revised form) by M. H. Gutknecht February 19, 2009; published electronically May 20, 2009. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.
http://www.siam.org/journals/simax/31-2/73239.html

†Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (greif@cs.ubc.ca, varah@cs.ubc.ca).

There are relatively few iterative algorithms for solving skew-symmetric systems. One such algorithm, of Huang, Wathen, and Li [12], has been shown in [13] to be equivalent to the CGNR algorithm [9, p. 105], by which, for skew-symmetric A, CG is applied to the symmetric positive definite system $-A^2x = -Ab$.

Specialized versions of CG may also be considered; e.g., the generalized conjugate gradient method of Concus & Golub [5]. But this algorithm and its variants cannot be used since they rely on solving in each iteration a system with the symmetric part of the originally given matrix; in our case the symmetric part is zero. One may ask whether it is possible to apply the algorithm to a shifted version of A, say, $\alpha I + A$ (which is not skew-symmetric for any $\alpha \neq 0$), and then take α to zero. Unfortunately, a small α causes instability, and taking it to zero results in a breakdown.

In this paper our goal is twofold: to present a complete framework for basic iterations, and to seek an effective preconditioning approach that can be combined into skew-symmetric solvers. We start by considering conjugate gradients for skew-symmetric systems. We use the Lanczos approach and show that in the skew-symmetric case there exists a backward stable block 2×2 decomposition for the tridiagonal matrix, which is key for the derivation of the scheme. We define a special kind of conjugacy, which we call *skew-A-conjugacy*, and set search directions that satisfy this relation. This direct derivation yields an algorithm that is equivalent to CGNE, whereby the symmetric positive definite system $-A^2y = b$ is solved, followed by setting $x = -Ay$.

In addition to the CG scheme, a minimum residual algorithm for skew-symmetric matrices is also of interest. Here there has been some recent interesting work on deriving basic (unpreconditioned) iterations for pure and shifted skew-symmetric systems; see [10, 13, 14]. We explain how the tridiagonal matrix arising in the Lanczos algorithm can be decomposed and used to obtain an iterative scheme tailored to skew-symmetric matrices.

Basic conjugate gradients and minimum residual iterations must be accompanied by preconditioning to make them practical. One question that arises is whether it is possible to derive effective preconditioning approaches that preserve the skew-symmetry. We address this central question and present an incomplete 2×2 block LDL^T decomposition.

The paper is structured as follows. In section 2, a Lanczos iteration for skew-symmetric matrices is given, and we show that the skew-symmetric tridiagonal matrix that arises in the process can be stably decomposed. The CG scheme that follows in section 3 is based on that decomposition and on a choice of special search directions and is equivalent to CGNE. Section 4 discusses the minimum residual approach. Section 5 is devoted to the issue of preconditioning. We close with sections 6 and 7, where we present a numerical example and draw some conclusions.

2. A Lanczos procedure and a stable decomposition.

Suppose A is an $n \times n$ matrix. Given a positive integer k and an initial unit vector v_1, the k-step Arnoldi iteration [1] for A generates a sequence of orthogonal vectors $\{v_j\}$ such that

$$AV_k = V_{k+1}H_{k+1,k}, \tag{2.1}$$

where $H_{k+1,k}$ is a $(k+1) \times k$ upper Hessenberg matrix and V_k is an $n \times k$ matrix whose orthonormal columns contain the vectors v_j, $j = 1, \ldots, k$. These vectors form an orthogonal basis for the Krylov subspace associated with A and the initial vector v_1.

Equation (2.1) readily implies $V_k^T AV_k = H_k$, where H_k is the square $k \times k$ matrix containing the first k rows of $H_{k+1,k}$. When A is symmetric, the upper Hessenberg matrix reduces to a tridiagonal matrix, which we denote by $T_{k+1,k}$, and the algorithm simplifies accordingly to the well-known Lanczos procedure.

2.1. Skew-Lanczos. It is straightforward to derive a Lanczos procedure for the skew-symmetric case, and such derivations and algorithms have been presented in a few papers [10, 13, 14]. The reduced matrix is tridiagonal and skew-symmetric. Thus, it has only k free coefficients:

$$T_{k+1,k} = \begin{pmatrix} 0 & \alpha_1 & & & & & \\ -\alpha_1 & 0 & \alpha_2 & & & & \\ & -\alpha_2 & 0 & \alpha_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & 0 & \alpha_{k-1} \\ & & & & & -\alpha_{k-1} & 0 \\ & & & & & & -\alpha_k \end{pmatrix}. \tag{2.2}$$

The construction procedure is given in Algorithm 1. Input for the algorithm is an initial vector b, and output is the matrix V_{k+1} and the parameters $\{\alpha_i\},\ i = 1, \ldots, k$, that form the tridiagonal matrix $T_{k+1,k}$ of (2.2). We call the algorithm *skew-Lanczos*. It is the same as Algorithm 1 in [10] with block size $s = 1$, Algorithm 3.2 in [13], and the algorithm in [14, section 3.2] with $\alpha = 0$.

Algorithm 1 Skew-Lanczos

set $v_1 = b/\|b\|_2,\ \alpha_0 = 0,\ v_0 = 0$
for $j = 1$ to k
  $z = Av_j - \alpha_{j-1}v_{j-1}$
  $\alpha_j = \|z\|_2$
  if $\alpha_j = 0$, quit
  $v_{j+1} = -z/\alpha_j$
end for

2.2. A backward stable block LDL^T decomposition for tridiagonal skew-symmetric matrices. The skew-Lanczos procedure generates a tridiagonal skew-symmetric matrix. If we are to derive an iterative method based on it, it is useful to ask what decompositions are available. We show in this section that this matrix admits a backward stable 2×2 block LDL^T decomposition. This is essential for developing the conjugate gradient iteration described in the next section. We will assume throughout that the dimension of the given matrix is even.

THEOREM 2.1. *Let $T_{2j} \in \mathbb{R}^{2j\times 2j}$ be a nonsingular tridiagonal skew-symmetric matrix given by*

$$T_{2j} = \begin{pmatrix} 0 & \alpha_1 & & & & & \\ -\alpha_1 & 0 & \alpha_2 & & & & \\ & -\alpha_2 & 0 & \alpha_3 & & & \\ & & -\alpha_3 & 0 & \alpha_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \\ & & & & -\alpha_{2j-2} & 0 & \alpha_{2j-1} \\ & & & & & -\alpha_{2j-1} & 0 \end{pmatrix}.$$

For $i = 1, \ldots, j$, denote

$$\widehat{L}_i = \begin{pmatrix} \ell_i & 0 \\ 0 & 0 \end{pmatrix},$$

with $\ell_i = -\frac{\alpha_{2i}}{\alpha_{2i-1}}$. Let I_2 denote a 2×2 identity matrix. Then

$$T_{2j} = L_{2j} D_{2j} L_{2j}^T,$$

where $L_{2j} \in \mathbb{R}^{2j \times 2j}$ is a 2×2 block unit lower bidiagonal matrix given by

$$L_{2j} = \begin{pmatrix} I_2 & & & & \\ \widehat{L}_1 & I_2 & & & \\ & \widehat{L}_2 & I_2 & & \\ & & \ddots & \ddots & \\ & & & \widehat{L}_{j-1} & I_2 \end{pmatrix},$$

and $D_{2j} \in \mathbb{R}^{2j \times 2j}$ is skew-symmetric block diagonal given by

$$D_{2j} = \left(\begin{array}{cc|cc|c|c|cc} 0 & \alpha_1 & & & & & & \\ -\alpha_1 & 0 & & & & & & \\ \hline & & 0 & \alpha_3 & & & & \\ & & -\alpha_3 & 0 & & & & \\ \hline & & & & \ddots & & & \\ \hline & & & & & \ddots & & \\ \hline & & & & & & 0 & \alpha_{2j-1} \\ & & & & & & -\alpha_{2j-1} & 0 \end{array}\right).$$

This decomposition is backward stable.

Proof. By its definition, the matrix L_{2j} is given by

$$L_{2j} = \left(\begin{array}{cc|cc|cc|c} 1 & 0 & & & & & \\ 0 & 1 & & & & & \\ \hline -\alpha_2/\alpha_1 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 1 & & & \\ \hline & & -\alpha_4/\alpha_3 & 0 & 1 & 0 & \\ & & 0 & 0 & 0 & 1 & \\ \hline & & & & \ddots & \ddots & \ddots \\ \hline & & & & & \ddots & \ddots \end{array}\right),$$

and it is straightforward to show that the decomposition stated in the theorem holds by construction, employing a block Gaussian elimination procedure. For backward stability, by [11, section 9.5] it is sufficient to show that

$$|T_{2j}| = |L_{2j}|\ |D_{2j}|\ |L_{2j}^T|.$$

Indeed,

$$|L_{2j}|\ |D_{2j}| = \left(\begin{array}{cc|cc|c} 0 & |\alpha_1| & & & \\ |\alpha_1| & 0 & & & \\ \hline 0 & |\alpha_2| & 0 & |\alpha_3| & \\ 0 & 0 & |\alpha_3| & 0 & \\ \hline & & & & \ddots \end{array}\right),$$

and hence

$$
\begin{aligned}
&|L_{2j}|\ |D_{2j}|\ |L_{2j}^T| \\
&= \left(\begin{array}{cc|cc|c}
0 & |\alpha_1| & & & \\
|\alpha_1| & 0 & & & \\
\hline
0 & |\alpha_2| & 0 & |\alpha_3| & \\
0 & 0 & |\alpha_3| & 0 & \\
\hline
 & & & & \ddots
\end{array}\right)
\left(\begin{array}{cc|cc|c}
1 & 0 & |\alpha_2/\alpha_1| & 0 & \\
0 & 1 & 0 & 0 & \\
\hline
 & & 1 & 0 & \\
 & & 0 & 1 & \\
\hline
 & & & & \ddots
\end{array}\right) \\
&= |T_{2j}|,
\end{aligned}
$$

as required. $\square$

3. Conjugate gradients. In what follows we adopt the notation used by Demmel [6, Chapter 6] in his elegant description of the derivation of CG for symmetric positive definite systems.

Consider the linear system

$$Ax = b,$$

where $A \in \mathbb{R}^{n \times n}$ is a large, sparse, and skew-symmetric nonsingular matrix. Note that nonsingularity implies that n must be even. Let $T_{2j} = V_{2j}^T A V_{2j}$ be the $(2j) \times (2j)$ skew-symmetric square tridiagonal matrix obtained by taking the first $2j$ rows of $T_{2j+1,2j}$ in skew-Lanczos (Algorithm 1). The columns of V_{2j} form an orthogonal basis for the space

$$\mathcal{K}^{2j}(A; b) = \operatorname{span}\{b, Ab, A^2 b, \dots, A^{2j-1} b\}, \tag{3.1}$$

and we have the following result.

THEOREM 3.1. *Given that A and T_{2j} are nonsingular, consider computing iterates of the form*

$$x_{2j} = V_{2j} T_{2j}^{-1} \left(\|b\|_2 e_1 \right).$$

Without loss of generality, suppose that $x_0 = 0$. Then the associated residuals $r_{2j} = b - Ax_{2j}$ satisfy a Galerkin condition: they are orthogonal to $\mathcal{K}^{2j}(A; b)$ defined in (3.1). Also, $v_{2j+1} = \frac{r_{2j}}{\|r_{2j}\|}$.

Proof. We proceed in an identical way to the proof for classical CG; see, for example, [6, Theorem 6.8]. We have

$$
\begin{aligned}
V_{2j}^T r_{2j} &= V_{2j}^T (b - Ax_{2j}) \\
&= V_{2j}^T b - V_{2j} A V_{2j} T_{2j}^{-1} \left(\|b\|_2 e_1 \right) \\
&= \|b\|_2 e_1 - T_{2j} T_{2j}^{-1} \left(\|b\|_2 e_1 \right) \\
&= 0.
\end{aligned}
$$

Finally, if $x_{2j} \in \mathcal{K}^{2j}(A; b)$, then $r_{2j} \in \mathcal{K}^{2j+1}(A; b)$, and hence by orthogonality we must have that v_{2j+1} can only be in the direction of r_{2j}. $\square$

From Theorem 2.1 we have $T_{2j} = L_{2j} D_{2j} L_{2j}^T$. Denote

$$\tilde{P}_{2j} = V_{2j} L_{2j}^{-T}, \qquad y_{2j} = D_{2j}^{-1} L_{2j}^{-1} \|b\|_2 e_1. \tag{3.2}$$

By these definitions, we have

$$x_{2j} = \tilde{P}_{2j}\, y_{2j}.$$

DEFINITION 3.2. *We say that a set of vectors* $\{s_i\}$ *is* skew-A-conjugate *with respect to a skew-symmetric matrix* A *if, for the matrix* S *containing* s_i *in its columns,* $S^T AS$ *is* 2×2 *block diagonal skew-symmetric.*

PROPOSITION 3.3. *The columns* $\{\tilde{p}_i\}, i = 1, \ldots, 2j,$ *of* $\tilde{P}_{2j}$ *defined in* (3.2) *are skew-A-conjugate.*

Proof. This is again a result that can be obtained similarly to the classical CG method:

$$\tilde{P}_{2j}^T A\tilde{P}_{2j} = L_{2j}^{-1}V_{2j}^T AV_{2j}L_{2j}^{-T} = L_{2j}^{-1}T_{2j}L_{2j}^{-T} = L_{2j}^{-1}(L_{2j}D_{2j}L_{2j}^T)L_{2j}^{-T} = D_{2j}.$$

Since D_{2j} is block diagonal skew-symmetric, the proof is complete. ☐

Let us write $V_{2j} = [v_1 \quad v_2 \quad \cdots \quad v_{2j-1} \quad v_{2j}] = [V_{2j-2} \quad v_{2j-1} \quad v_{2j}]$. We can derive the $\tilde{p}$-recurrences from $\tilde{P}_{2j}L_{2j}^T = V_{2j}$:

$$[\tilde{p}_{2j+1} \quad \tilde{p}_{2j+2}] = [v_{2j+1} \quad v_{2j+2}] - [\tilde{p}_{2j-1} \quad \tilde{p}_{2j}] \begin{pmatrix} \ell_j & 0 \\ 0 & 0 \end{pmatrix}.$$

So, the $\tilde{p}$-vectors satisfy the recurrence relations

$$\begin{cases} \tilde{p}_{2j+1} = v_{2j+1} - \ell_j \tilde{p}_{2j-1}, \\ \tilde{p}_{2j+2} = v_{2j+2}. \end{cases}$$

The v-vectors are available from the skew-Lanczos procedure. In particular, from Algorithm 1 we have, for any k, $-\alpha_k v_{k+1} = Av_k - \alpha_{k-1}v_{k-1}$.

Suppose we have constructed a $(2j)$-dimensional orthogonal basis for the Krylov subspace. By Theorem 3.1, $v_{2j+1} = \frac{r_{2j}}{\|r_{2j}\|}$, and it follows that

$$-\alpha_{2j+1}v_{2j+2} = Ar_{2j}/\|r_{2j}\| - \alpha_{2j}v_{2j}.$$

Since $\ell_j = -\alpha_{2j}/\alpha_{2j-1}$ and $v_{2j} = \tilde{p}_{2j}$, we obtain the relations

$$\begin{cases} \tilde{p}_{2j+1} = r_{2j}/\|r_{2j}\| + \frac{\alpha_{2j}}{\alpha_{2j-1}}\tilde{p}_{2j-1}, \\ \tilde{p}_{2j+2} = (Ar_{2j}/\|r_{2j}\| - \alpha_{2j}\tilde{p}_{2j})/(-\alpha_{2j+1}). \end{cases}$$

These recurrences can be simplified by normalization. Set

$$p_{2j+1} = \|r_{2j}\| \cdot \tilde{p}_{2j+1},$$

and define

$$\mu_j = \frac{\alpha_{2j}\|r_{2j}\|}{\alpha_{2j-1}\|r_{2j-2}\|}.$$

This gives

$$p_{2j+1} = r_{2j} - \mu_j p_{2j-1}.$$

Setting $p_{2j+2} = -\alpha_{2j+1}\|r_{2j}\|\tilde{p}_{2j+2}$ gives

$$p_{2j+2} = Ar_{2j} + \frac{\alpha_{2j}\|r_{2j}\|}{\alpha_{2j-1}\|r_{2j-2}\|}p_{2j} = Ar_{2j} - \mu_j p_{2j}. \tag{3.3}$$

To find μ_j we multiply (3.3) by p_{2j}^T on the left and use the fact that p_{2j} is in the direction of $\tilde{p}_{2j}$ and hence of v_{2j}. Thus, p_{2j+2} is orthogonal to p_{2j}, and we get

$$\mu_j = \frac{p_{2j}^T A r_{2j}}{p_{2j}^T p_{2j}}.$$

We will see soon that once the other recurrence relations are derived we can obtain an alternative expression for μ_j that does not require a multiplication with the matrix A. This is similar to the situation for classical CG; see [6].

We now proceed to find a short recurrence relation for the vectors y_{2j}. Here we have the following useful result.

THEOREM 3.4. *The vectors y_{2j} have the following properties:*

1. *For a given $j > 1$, the first $2j-2$ elements of the $(2j)$-vector y_{2j} are equal componentwise to the elements of the $(2j-2)$-vector y_{2j-2}.*
2. *The odd-indexed elements of y_{2j} are identically equal to* 0.

Remark. While property 1 holds for classical CG for symmetric positive definite matrices, property 2 seems to be unique to skew-symmetric matrices and allows for simplification.

Proof. Denote

$$z_{2j} = L_{2j}^{-1}(\|b\|e_1).$$

From the block structure of the unit lower triangular matrix L_{2j} it follows that

$$L_{2j}^{-1} = \left(\begin{array}{c|c} L_{2j-2}^{-1} & 0 \\ \hline * & I_2 \end{array}\right),$$

where $*$ denotes a value that is not to be used below. The first $2j-2$ elements of z_{2j} are thus equal to z_{2j-2} and similarly for y_{2j} and y_{2j-2}.

Let us write $z_{2j} = \begin{pmatrix} z_{2j-2} \\ \zeta_{2j-1} \\ \zeta_{2j} \end{pmatrix}$. Then, since $L_{2j}z_{2j} = \|b\|e_1$, we see that

$$\left(\begin{array}{c|c} L_{2j-2} & 0 \\ \hline l_{2j} & I_2 \end{array}\right) \begin{pmatrix} z_{2j-2} \\ \zeta_{2j-1} \\ \zeta_{2j} \end{pmatrix} = \begin{pmatrix} \|b\| \\ 0 \\ \vdots \end{pmatrix},$$

where $l_{2j} = [0 \ \cdots \ 0 \ \widehat{L}_{j-1}]$ is $2 \times (k-2)$. Writing out the last two equations, we have

$$\begin{pmatrix} \ell_{2j} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_o \\ z_e \end{pmatrix} + \begin{pmatrix} \zeta_{2j-1} \\ \zeta_{2j} \end{pmatrix} = 0,$$

where z_o and z_e are the last two elements of z_{2j}. From this it readily follows that $\zeta_{2j} = 0$. Now, $y_{2j} = D_{2j}^{-1}z_{2j}$. We write

$$y_{2j} = \begin{pmatrix} y_{2j-2} \\ \eta_{2j-1} \\ \eta_{2j} \end{pmatrix}$$

and get

$$\begin{pmatrix} \eta_{2j-1} \\ \eta_{2j} \end{pmatrix} = \begin{pmatrix} 0 & \times \\ \times & 0 \end{pmatrix} \begin{pmatrix} \zeta_{2j-1} \\ \zeta_{2j} \end{pmatrix},$$

so $\eta_{2j-1} = 0$. □

The result of Theorem 3.4 allows us to simplify the recurrence relations for x_{2j}. Instead of $x_{2j+2} = x_{2j} + \eta_{2j+1}\tilde{p}_{2j+1} + \eta_{2j+2}\tilde{p}_{2j+2}$ we in fact have

$$\begin{aligned} x_{2j+2} &= x_{2j} + \eta_{2j+2}\tilde{p}_{2j+2} \\ &= x_{2j} + \nu_{2j} p_{2j+2}, \end{aligned}$$

where we define

$$\nu_{2j} = -\frac{\eta_{2j+2}}{\alpha_{2j+1}\|r_{2j}\|}.$$

The residual vectors satisfy

$$r_{2j+2} = b - Ax_{2j+2} = r_{2j} - \nu_{2j+2} A p_{2j+2}. \tag{3.4}$$

To find ν_{2j+2}, we multiply (3.4) by r_{2j}^T from the left and use the fact that even-indexed residuals are orthogonal to each other, since the orthogonal set of vectors v_{2j+1} are in the direction of r_{2j}. Thus, we have

$$\nu_{2j+2} = \frac{r_{2j}^T r_{2j}}{r_{2j}^T A p_{2j+2}}.$$

This expression can be simplified. Since p_{2j+1} and $\tilde{p}_{2j+1}$ are in the same direction, we have $p_{2j+2}^T A p_{2j} = 0$. Thus, by (3.3) $p_{2j+2}^T p_{2j+2} = p_{2j+2}^T A r_{2j}$, from which it follows (using the skew-symmetry of A) that

$$\nu_{2j+2} = -\frac{r_{2j}^T r_{2j}}{p_{2j+2}^T p_{2j+2}}.$$

We now go back to the expression for μ_j and show that in fact it does not require multiplying by A. If we multiply (3.4) by r_{2j+2}^T on the left and use orthogonality, we get

$$r_{2j+2}^T r_{2j+2} = -\nu_{2j+2} r_{2j+2}^T A p_{2j+2} = \nu_{2j+2} p_{2j+2}^T A r_{2j+2}.$$

By equating the last two expressions for ν_{2j+2} we get

$$\mu_j = -\frac{r_{2j}^T r_{2j}}{r_{2j-2}^T r_{2j-2}}. \tag{3.5}$$

We can now collect these recurrence relations into a short algorithm. It is possible to further simplify the algorithm in terms of indexing: since the odd-indexed p vectors need not be computed due to Theorem 3.4, we can in fact change $2j$ to j everywhere. Note that a single iteration will involve *two* matrix-vector products. We show the sequence of steps in Algorithm 2, with the sign of μ_j changed. In the algorithm we keep the initial guess at zero for simplicity and consistency with the discussion so far.

Examining this scheme reveals that it is equivalent to CGNE [9, p. 105], which is applicable to general matrices and amounts to solving $AA^T y = b$ and then setting $x = A^T y$. In the skew-symmetric case it is thus equivalent to solving $-A^2 y = b$ and setting $x = -Ay$.

Convergence analysis for the scheme can be directly drawn from the fact that it is equivalent to CGNE. Recall that our scheme proceeds from skew-Lanczos, with

V_{2j} an orthogonal basis for the Krylov space $\mathcal{K}^{2j}$ defined in (3.1). We have $x_2 \in \text{span}\{Ab\}$, $x_4 \in \text{span}\{Ab, A^3b\}, \ldots, x_{2j} \in \text{span}\{Ab, \ldots, A^{2j-1}b\}$. It follows that $x_{2j} \in \text{range}(A\mathcal{K}^{2j})$, and so there exists a vector u_{2k} such that $x_{2j} = Au_{2j}$.

Consider now the solution x of $Ax = b$, with A skew-symmetric and nonsingular. The skew-symmetry of A implies $x^T b = x^T Ax = 0$. Similarly, $x^T A^2 b = x^T A^4 b = \cdots = 0$, so $x \in \text{span } \{Ab, A^3b, \ldots\}$. From this we conclude that there is a vector u such that $x = -Au$ and $-A^2u = b$. We thus have

$$\|x_{2j} - x\|_2^2 = (x_{2j} - x)^T (x_{2j} - x) = (u_{2j} - u)^T (-A^2)(u_{2j} - u) = \|u_{2j} - u\|_{-A^2}^2.$$

Algorithm 2 Conjugate gradients for a skew-symmetric system

$x_0 = 0, \quad r_0 = b, \quad \mu_0 = 0, \quad p_0 = 0$
for $j = 0, 1, \ldots$
  if $j \geq 1$, then $\mu_j = \frac{r_j^T r_j}{r_{j-1}^T r_{j-1}}$
  $p_{j+1} = Ar_j + \mu_j p_j$
  $\nu_j = -\frac{r_j^T r_j}{p_{j+1}^T p_{j+1}}$
  $x_{j+1} = x_j + \nu_j p_{j+1}$
  $r_{j+1} = r_j - \nu_j A p_{j+1}$
end for

From this it follows, using standard convergence analysis, that

$$\|x_{2j} - x\|_2 \leq 2 \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^{2j} \|x_0 - x\|_2.$$

As expected, the factor of $\sqrt{\kappa}$ for classical CG is replaced here by κ.

It also follows that if A has only $2k$ distinct eigenvalues $\lambda_i = \pm \imath \alpha_i, \ i = 1, \ldots, k$, the minimal polynomial is

$$p_{2k}(z) = (z^2 + \alpha_1^2)(z^2 + \alpha_2^2) \cdots (z^2 + \alpha_k^2),$$

and hence the scheme will converge within $2k$ iterations in exact arithmetic.

4. Minimum residuals for skew-symmetric systems. MINRES [15] seeks in every iteration to compute the least squares solution within the Krylov subspace; the iterate x_k minimizes $\|b - Ay\|_2$ over all vectors $y \in x_0 + \mathcal{K}^k(A; r_0)$.

Recently, a few papers [10, 13, 14] have introduced basic (unpreconditioned) minimum residual iterations for shifted and pure skew-symmetric systems. Specifically, the MRS^3 scheme in [13] is elegantly derived in a complete and comprehensive fashion. What is not discussed in these papers is how preconditioners can be incorporated into the skew-symmetric solvers. We will later describe a preconditioned Lanczos algorithm for skew-symmetric matrices; this is the basis for a preconditioned minimum residual iteration.

Without loss of generality, we assume that $x_0 = 0$ and hence $r_0 = b$. Recall that after skew-Lanczos we have $AV_k = V_{k+1} T_{k+1,k}$ with $T_{k+1,k}$ skew-symmetric and V_k an orthogonal basis of $\mathcal{K}^k(A; r_0)$. Hence the MINRES solution can be written as $x_k = V_k y_k$, and we have

$$\begin{aligned} b - Ax_k &= b - AV_k y_k \\ &= b - V_{k+1} T_{k+1,k} y_k. \end{aligned}$$

Using the orthogonality of V_{k+1} and the fact that by construction the first column is given by $v_1 = b/\|b\|_2$, the least squares problem turns into

$$\min_y \|\rho e_1 - T_{k+1,k} y\|_2,$$

where $\rho = \|b\|_2$. Here e_1 is the first standard basis vector of size $k+1$. This is well known and can be found in many numerical linear algebra textbooks; see, e.g., [9].

Below we will refer both to the full QR factorization of $T_{k+1,k}$ and to its economy size factorization; the corresponding Q factors are to be denoted by Q_{k+1} and $\widehat{Q}_{k+1}$, respectively, and the square upper triangular factor is to be denoted by R_k. That is,

$$T_{k+1,k} = Q_{k+1} \begin{pmatrix} R_k \\ 0 \end{pmatrix} \equiv \widehat{Q}_{k+1} R_k,$$

where $Q_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$, $\widehat{Q}_{k+1} \in \mathbb{R}^{(k+1)\times k}$, and $R_k \in \mathbb{R}^{k\times k}$. The matrix $\widehat{Q}_{k+1}$ consists of the first k rows of Q_{k+1}.

The minimum residual solution is given by

$$x_k = (V_k R_k^{-1})(\widehat{Q}_{k+1}^T \rho e_1).$$

We set

$$W_k = V_k R_k^{-1}\ , \quad z_k = \widehat{Q}_{k+1}^T \rho e_1^{(k+1)}$$

and seek short recurrence relations for W_k and z_k, exploiting the skew-symmetry of A and $T_{k+1,k}$. To accomplish this, explicit expressions for the Q and R factors of $T_{k+1,k}$ will be derived. The kth iterate is then $x_k = W_k z_k$.

4.1. The QR factorization of $T_{k+1,k}$. As done in [15] for symmetric matrices and in [13] for shifted skew-symmetric matrices, to obtain the QR factorization of $T_{k+1,k}$ we use a sequence of Givens rotations. To illustrate the procedure and point out how the skew-symmetric case works, we perform the first few steps of the factorization in detail.

Starting off with $T_{2,1} = \begin{pmatrix} 0 \\ -\alpha_1 \end{pmatrix}$, the skew-Lanczos process (Algorithm 1) gives us $Av_1 = (v_1,\ v_2)\ T_{2,1}$. The QR factorization is thus trivially given by $T_{2,1} = Q_2 \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, where $Q_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $R_1 = \begin{pmatrix} \alpha_1 \end{pmatrix}$. Now, consider the 3×2 matrix $T_{3,2}$ that is generated after the first $k = 2$ steps of skew-Lanczos:

$$T_{3,2} = \begin{pmatrix} 0 & \alpha_1 \\ -\alpha_1 & 0 \\ 0 & -\alpha_2 \end{pmatrix}.$$

The rotation matrices are given (in transposed form) by

$$G_1^T = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}; \quad G_2^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_2 & -s_2 \\ 0 & s_2 & c_2 \end{pmatrix}; \quad c_2 = \frac{\alpha_1}{\sqrt{\alpha_1^2 + \alpha_2^2}}; \quad s_2 = \frac{\alpha_2}{\sqrt{\alpha_1^2 + \alpha_2^2}}.$$

We define $Q_3 = G_1 G_2$ and have

$$Q_3^T T_{3,2} = G_2^T G_1^T T_{3,2} = \left(\begin{array}{cc} \alpha_1 & 0 \\ 0 & \sqrt{\alpha_1^2 + \alpha_2^2} \\ \hline 0 & 0 \end{array}\right) \equiv \begin{pmatrix} R_2 \\ 0 \end{pmatrix}. \tag{4.1}$$

The 2×2 upper triangular matrix R_2 is the R factor of the QR factorization of $T_{3,2}$, and $Q_3 = G_1 G_2$ is the Q factor.

The procedure can now be repeated. Assuming that k is even, in every step we define two new $(k+1) \times (k+1)$ Givens rotation matrices G_{k-1} and G_k and apply the corresponding pair of rotations. The resulting QR factorization is given by

$$T_{k+1,k} = Q_{k+1} \begin{pmatrix} R_k \\ 0 \end{pmatrix},$$

where $Q_{k+1} = \prod_{j=1}^{k} G_j$ is $(k+1) \times (k+1)$ but need not be formed explicitly, and R_k is $k \times k$; 0 in the above equation represents a single row of k zero elements. For a given $k = 2j$, the diagonal elements of R_k are given as follows:

$$r_{i,i} = \begin{cases} \alpha_i, & i = 1, 3, 5, \ldots, k-1, \\ \sqrt{(\alpha_{i-1} c_{i-2})^2 + \alpha_i^2}, & i = 2, 4, 6, \ldots, k, \end{cases} \tag{4.2}$$

where in the second equation of (4.2) $c_0 \equiv 1$. The nonzero off-diagonal elements of R_k are

$$r_{i,i+2} = \begin{cases} -\alpha_{i+1}, & i = 1, 3, 5, \ldots, k-3, \\ -\alpha_{i+1} s_i, & i = 2, 4, 6, \ldots, k-2. \end{cases} \tag{4.3}$$

Note that the only two nonzero diagonals of R_k are the main diagonal and the second superdiagonal. This is different from the situation in the symmetric case, in which the main diagonal and the two superdiagonals immediately above it are nonzero.

4.2. Minimum residual iterates. Having established the structure of the QR factorization of $T_{k+1,k}$, we can now formulate the algorithm. We have $x_1 = (V_1 R_1^{-1}) \widehat{Q}_2^T \rho e_1 \equiv W_1 z_1$, where $W_1 = \frac{v_1}{\alpha_1} = \frac{b}{\alpha_1 \rho} \in \mathbb{R}^{n \times 1}$ and e_1 is of length 2. But since $z_1 = (0, -1)\rho\binom{1}{0} = 0$ we conclude that $x_1 = 0$. As we shall see soon, odd-indexed iterations are identically zero, and progressing toward convergence requires performing two matrix-vector products at a time. This is thus the same situation as that for the conjugate gradient scheme of section 3.

Next we have

$$x_2 = \left((v_1,\ v_2)\ R_2^{-1} \right) \left(\widehat{Q}_3^T \rho e_1 \right) \equiv W_2\, z_2.$$

We can see that $W_2 = \left(\frac{v_1}{\alpha_1}\ \frac{v_2}{\sqrt{\alpha_1^2 + \alpha_2^2}} \right)$ and $z_2 = \binom{0}{\rho c_2}$. This result can now be generalized into the following theorem.

THEOREM 4.1. *At the kth iterate (k even) the first $k-2$ elements of z_k are identical to those of z_{k-2}. Furthermore, the odd-indexed elements of z_k are zero. The even-indexed elements of z_k are given by*

$$\zeta_i = \rho c_i \cdot \prod_{j=2,4,\ldots,i-2} s_j\ , \qquad i = 2,\ 4,\ \ldots.$$

(For $i = 2$ the product is defined to be 1.)

Proof. We have already shown that the first element of z_2 is zero and its second element is $\zeta_2 = \rho c_2$, as required. The vector $Q_3^T \rho e_1$ has the components of z_2 as its first two elements, and its third element is equal to ρs_2.

Now, by induction, given k even, suppose that z_{k-2} is as stated in the theorem $z_{k-2} = (0, \zeta_2, 0, \zeta_4, \ldots, 0, \zeta_{k-2})^T$, and suppose also that $Q_{k-1}^T \rho e_1 = [z_{k-2}; \tilde{z}_{k-1}]$, where $\tilde{z}_{k-1} = \rho \prod_{j=2,4,\ldots,k-2} s_j$. The next two Givens rotations for forming the QR factorization of $T_{k+1,k}$ are $(k+1) \times (k+1)$, as follows. G_{k-1}^T is the identity matrix, except its $[k-1:k, k-1:k]$ block is given by $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. The matrix G_k^T is given by the identity matrix, except its $[k:k+1, k:k+1]$ block is $\left(\begin{smallmatrix} c_k & -s_k \\ s_k & c_k \end{smallmatrix}\right)$. Take $[z_{k-2}; \tilde{z}_{k-1}]$ and pad it by two zeros so as to match the required size for the next pair of rotations. After multiplying by G_{k-1}^T we have that $\tilde{z}_{k-1}$ is now placed as the kth element of the resulting vector, and the $(k-1)$st element becomes zero. Since multiplying by G_k^T changes only the last two elements of the vector, which is of size $k+1$ after padding, it follows that the first $k-1$ elements are not changed; in particular, the $(k-1)$st element continues to be zero. The last two elements are equal to $[\tilde{z}_{k-1}; 0]$ before the rotation. Hence, after G_k^T is applied the kth element must be ζ_k, as defined in the statement of the theorem. The $(k+1)$st element must be $s_k \tilde{z}_{k-1}$ (which, after k is incremented, will be equal to $\tilde{z}_k$, as required by the induction), but notice that since $\widehat{Q}_{k+1}$ is the matrix consisting of the first k columns of Q_{k+1}, it follows that this element is not part of z_k and will be used only in the next pair of iterates if and when they are performed. ◻

Next, we consider the recurrence relation for the W_k matrices. Consider $V_k = W_k R_k$. Column by column, we have

$$v_k = w_{k-2} r_{k-2,k} + w_k r_{k,k}.$$

This is different than the situation for symmetric matrices, since here the R factor has only two rather than three nonzero elements per row. We can now use the formulas we have for the elements of R_k, given in (4.2)–(4.3). But before we proceed, we make an observation that allows us to save computational work. Since $x_k = W_k z_k$ and z_k has all its odd-indexed elements zero by Theorem 4.1, only the even-indexed W_k play a role in the iteration. Therefore, we need only even-indexed columns, w_{2j}, at every step.

By skew-Lanczos the initial condition is $w_2 = \frac{v_2}{\sqrt{\alpha_1^2 + \alpha_2^2}}$, and the recurrence relation is given by

$$w_{2j} = \frac{v_{2j} + \alpha_{2j-1} s_{2j-2} w_{2j-2}}{\sqrt{(\alpha_{2j-1} c_{2j-2})^2 + \alpha_{2j}^2}}, \qquad j = 2, 3, 4, \ldots.$$

The solution is given by

$$x_k = w_2 \zeta_2 + w_4 \zeta_4 + \cdots + w_k \zeta_k.$$

5. Preconditioning. In this section we develop preconditioned iterations. We consider the system

$$M_1^{-1} A M_2^{-1} (M_2 x) = M_1^{-1} b \qquad \longleftrightarrow \qquad \widehat{A}\widehat{x} = \widehat{b}. \tag{5.1}$$

For conjugate gradients, given the equivalence to CGNE, a preconditioned approach may work well if $\widehat{A}\widehat{A}^T$ is easy to invert, and this gives rise to various possibilities, for example, $\widehat{A} = M_1^{-1} A M_2^{-1} \approx Q$ with Q orthogonal. Here one may consider, for example, applying an incomplete LQ factorization, and then taking $M_1 = L$ and $M_2 = I$. This yields computationally efficient iterations, which may rapidly converge.

Note, however, that $\widehat{A}$ is not necessarily skew-symmetric in this case, and hence the skew-symmetric solvers we have discussed thus far cannot be directly utilized.

We will pursue below a preconditioning approach that preserves skew-symmetry, working with A rather than with the symmetric positive definite matrix $-A^2$. Dealing with A rather than $-A^2$ may give rise to a rich class of preconditioners, since every preconditioner that keeps the system associated with A skew-symmetric must necessarily keep the system associated with $-A^2$ symmetric positive definite, but the opposite is not true. Also, directly dealing with A may be advantageous when the problem comes from an application where the operator A has properties that can be exploited; for example, if it comes from a discretized differential equation. It is possible that while A can be interpreted in terms of the underlying application, the positive definite matrix $-A^2$ (which is less sparse and worse conditioned) may not have a natural and easy to exploit interpretation.

The rate of convergence is governed by the spectrum of the preconditioned matrix; clustered eigenvalues will lead to faster convergence. Hence, a stated goal in the development of preconditioners is to accomplish a favorable spectral structure.

5.1. A preconditioned conjugate gradient iteration. Given (5.1), we seek M_1 and M_2 such that the preconditioned system is also skew-symmetric, and we can derive a preconditioned iteration as follows. Again we assume without loss of generality that the initial guess is zero and define $\widehat{x}_0 = \widehat{p}_0 = 0,\ \widehat{r}_0 = \widehat{b} = M_1^{-1}b,\ \mu_0 = 0$. Following [6, p. 317], we use

$$p_j = M_2^{-1}\widehat{p}_j, \qquad x_j = M_2^{-1}\widehat{x}_j, \qquad r_j = M_1\widehat{r}_j.$$

For notational convenience, we will eventually eliminate the quantities with the hats. To this end, set $y_j = \widehat{r}_j = M_1^{-1}r_j$. We have $x_0 = p_0 = 0, r_0 = b, \mu_0 = 0$, and

$$M_2p_{j+1} = M_1^{-1}AM_2^{-1}y_j + \widehat{\mu}_jM_2p_j.$$

So, from this it follows that $M_2(p_{j+1} - \widehat{\mu}_jp_j) = v_j$, where $M_1v_j = Az_j$, with $z_j = M_2^{-1}y_j$. For $\widehat{\mu}_j$ we have

$$\widehat{\mu}_j = \frac{y_j^Ty_j}{y_{j-1}^Ty_{j-1}},$$

and the solution and residual are computed by $x_{j+1} = x_j + \widehat{\nu}_jp_{j+1}$ and $r_{j+1} = r_j - \widehat{\nu}_jAp_{j+1}$. Finally,

$$\widehat{\nu}_j = -\frac{y_j^Ty_j}{\widehat{p}_{j+1}^T\widehat{p}_{j+1}} = -\frac{y_j^Ty_j}{(M_2p_{j+1})^TM_2p_{j+1}},$$

which is updated after updating p_j:

$$p_{j+1} = \widehat{\mu}_jp_j + M_2^{-1}v_j.$$

The iteration is given in Algorithm 3, where we have replaced $\widehat{\mu}_j$ and $\widehat{\nu}_j$ by μ_j and ν_j, respectively. The algorithm is now minimizing $\|\widehat{e}_k\|_2 = \|M_2e_k\|_2$.

Note that we need not impose a skew-symmetry requirement on M_1 and M_2. For example, taking M_1 nonsingular and $M_2 = M_1^T$ gives rise to a rich class of possible preconditioners, including incomplete factorizations of the form discussed in section 5.3.

Algorithm 3 Preconditioned conjugate gradients for skew-symmetric systems

$x_0 = p_0 = w_0 = 0, \quad r_0 = b, \quad y_0 = M_1^{-1} r_0, \quad \mu_0 = 0$
for $j = 0, 1, \ldots$
  if $j \geq 1$, $\mu_j = \frac{y_j^T y_j}{y_{j-1}^T y_{j-1}}$
  $v_j = M_1^{-1} A M_2^{-1} y_j$
  $p_{j+1} = M_2^{-1} v_j + \mu_j p_j$
  $w_{j+1} = v_j + \mu_j w_j$
  $\nu_j = -\frac{y_j^T y_j}{w_{j+1}^T w_{j+1}}$
  $x_{j+1} = x_j + \nu_j p_{j+1}$
  $r_{j+1} = r_j - \nu_j A p_{j+1}$
  $y_{j+1} = M_1^{-1} r_{j+1}$
end for

5.2. Preconditioned skew-Lanczos and minimum residuals. Similarly to what we did for CG, for MINRES we will replace $Ax = b$ (with A skew-symmetric) with the preconditioned system (5.1), taking $M_2 = M_1^T$:

$$M_1^{-1} A M_1^{-T} \widehat{x} = \widehat{b},$$

where $\widehat{x} = M_1^T x$ and $\widehat{b} = M_1^{-1} b$. The preconditioned minimum residual algorithm can be derived by means similar to the standard derivation for symmetric matrices; see, for example, [9, pp. 121–122]. A preconditioned version of the Lanczos procedure is key for developing a preconditioned MINRES iteration.

We now derive a preconditioned skew-Lanczos algorithm. We denote the preconditioner by M, i.e., $M = M_1 M_1^T$. Given $\widehat{v}_1 = \widehat{b}/\|\widehat{b}\|_2$, the first vector in the Lanczos procedure, we have the initial relation $M_1^{-1} A M_1^{-T} \widehat{v}_1 = -\alpha_1 \widehat{v}_2$. Define $v_1 = M_1 \widehat{v}_1$ and $w_1 = M^{-1} v_1$. Notice that $v_1 = M_1 \widehat{b}/\|\widehat{b}\| = b/\|M_1^{-1} b\|$ does not have unit norm. Following skew-Lanczos, we see that $\alpha_1 = \|M_1^{-1} A w_1\|$. It is convenient to define $z_1 = A w_1$ and $y_1 = M^{-1} z_1$; we then have $\alpha_1 = \sqrt{z_1^T M^{-1} z_1} = \sqrt{z_1^T y_1}$, $v_2 = -\frac{A w_1}{\alpha_1} = -\frac{z_1}{\alpha_1}$, and $w_2 = M^{-1} v_2 = -\frac{y_1}{\alpha_1}$. Note that M is inverted only once, in the computation of y_1, and then α_1, v_2, and w_2 are available without any additional inversion. Next, set $z_2 = A w_2 - \alpha_1 v_1$ and $y_2 = M^{-1} z_2$, which gives $\alpha_2 = \sqrt{z_2^T y_2}$, $v_3 = -\frac{z_2}{\alpha_2}$, and $w_3 = -\frac{y_2}{\alpha_2}$.

From here, the general procedure becomes evident. Given $v_0 = 0$, $\alpha_0 = 0$, $v_1 = b/\|M_1^{-1} b\|$, $w_1 = M^{-1} v_1$, the k-step preconditioned Lanczos procedure is based on the following main steps for $j = 1$ to k:

1. Compute $z_j = A w_j - \alpha_{j-1} v_{j-1}$.
2. Set $y_j = M^{-1} z_j$.
3. Set $\alpha_j = \sqrt{z_j^T y_j}$.
4. Set $v_{j+1} = -\frac{z_j}{\alpha_j}$ and $w_{j+1} = -\frac{y_j}{\alpha_j}$.

The preconditioned minimum residual solver can now be implemented by incorporating the new quantities of the preconditioned Lanczos procedure. We follow the procedure laid out in section 4, using Givens rotations to obtain a QR decomposition of the tridiagonal matrix, and we get iterates that minimize the residuals for the preconditioned system, as desired. Every iteration involves a preconditioner solve. Notice that $\{v_j\}$ are M^{-1}-orthogonal; it is $\widehat{v}_j$ that are orthonormal, and they are available without additional preconditioner inversions.

5.3. An incomplete block LDL^T factorization. Since the diagonal of A is zero, adopting a block incomplete factorization with blocks of size (at least) 2×2 seems a necessity. Bunch's decomposition [3] for skew-symmetric matrices forms

$$PAP^T = LDL^T,$$

where D consists of 2×2 diagonal blocks of the form

$$\begin{pmatrix} 0 & \alpha_i \\ -\alpha_i & 0 \end{pmatrix}$$

and L is unit block lower triangular. The permutation matrix P represents pivoting steps to improve stability. To maintain sparsity, we can apply this procedure in an incomplete fashion. As usual, a block $ILDL^T(0)$ factorization corresponds to restricting the factor L to have the same block structure of A. The matrix D is 2×2 block diagonal and is skew-symmetric. Note that a 2×2 block Gaussian Elimination step maintains the skew-symmetry, since the local 2×2 Schur complements are necessarily skew-symmetric.

During the kth elimination step we form a 2×2 block L_{ik} when the corresponding block $A_{ik} \neq 0$, and we perform the elimination on block (i, j) only when $A_{ij} \neq 0$ or when $i = j$ (the diagonal blocks).

A difficulty with an incomplete factorization of the form LDL^T with D skew-symmetric and L unit lower triangular is that it does not immediately fall in the category of preconditioners that can easily preserve the skew-symmetry of the (preconditioned) operator. The matrix

$$J_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

does have a square root matrix: $J_2^{1/2} = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$. However, this matrix is neither symmetric nor skew-symmetric. Instead, we choose a simple symmetric approximation of $J_2^{1/2}$, which preserves the skew-symmetry of the preconditioned matrix. To accomplish this, we approximate D by 2×2 blocks of the form J_2 or $-J_2$ by left and right diagonal scaling with $\widehat{D}^{-1}$, where

$$\widehat{D} = \text{diag}\begin{pmatrix} |\alpha_i|^{1/2} & 0 \\ 0 & |\alpha_i|^{1/2} \end{pmatrix}.$$

Thus, we set $M_1 = L\widehat{D}$, and we have

$$M_1^{-1}(PAP^T)M_1^{-T} \approx \text{diag}(\pm J_2),$$

where the matrix on the right is 2×2 block diagonal comprised of $n/2$ blocks, each either J_2 or $-J_2$. Notice that M_1 is triangular, and hence it easy to solve systems involving the preconditioner $M_1 M_1^T$.

Since $\text{diag}(\pm J_2)$ has only two eigenvalues, convergence improves when our approximation does, and in the limiting case we converge within two iterations.

The decomposition can be implemented in a direct way. Consider the first step.

We have

$$A = \left(\begin{array}{cc|ccc} 0 & -a_{21} & -a_{31} & \dots & -a_{n1} \\ a_{21} & 0 & -a_{32} & \dots & -a_{n2} \\ \hline \vdots & \vdots & & & \\ \vdots & \vdots & & A_2 & \\ a_{n1} & a_{n2} & & & \end{array} \right).$$

The elements of the first two columns of L are the multipliers $\{a_{k1}/a_{21},\ a_{k2}/a_{21}\}$. To keep these elements of reasonable size, various forms of pivoting can be used. In [3] Bunch proposed interchanging rows 2 and k (and columns 2 and k) if $|a_{k1}| = \max_j\{|a_{j1}|, |a_{j2}|\}$, with an obvious modification if the maximum is in the second column. Although this can work well in practice, it does not guarantee $|\ell_{j2}| \leq 1$ since after the column interchange, the elements of column 2 may well be larger in magnitude than a_{k1}. Alternatively, one can use *rook* pivoting [4, 11], where alternate row and column pivoting continues until an element is found which is the largest in its row and column. This generally takes only a few steps (our maximum in our experiments was three), and the method has very good stability properties [4] and guarantees $|\ell_{j1}|, |\ell_{j2}| \leq 1$.

6. A numerical example. This section is brief; we limit ourselves to a simple but representative example that illustrates the convergence behavior of the basic schemes and their preconditioned counterparts. Convergence graphs are given in Figure 6.1. Here the matrix is $4,096 \times 4,096$ and is the skew-symmetric part of the discrete convection-diffusion matrix, derived by the centered finite difference scheme on the unit square, 64×64 grid, with Dirichlet boundary conditions. The mesh Reynolds numbers are 0.5 and 0.6. We precondition with the incomplete 2×2 block skew-LDL^T decomposition described in section 5.3. As evident from the graphs, preconditioning is very effective. As expected, the convergence of the residual for the minimum residual scheme is monotonic and smooth, as is the convergence of the error for the conjugate gradient algorithm. The performance of the two schemes is very similar in this case.

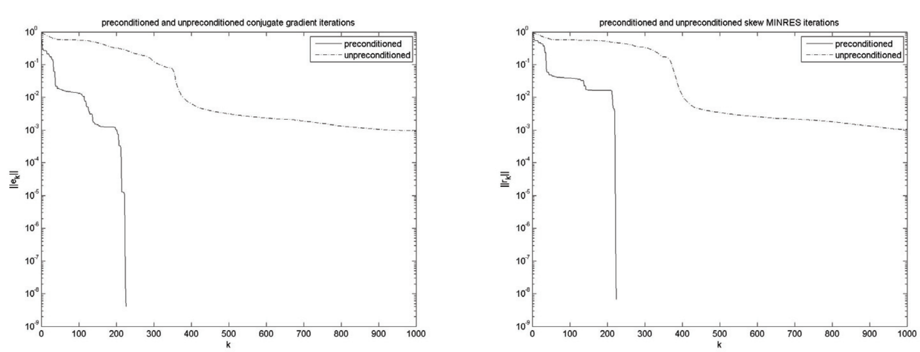

FIG. 6.1. *Convergence behavior of conjugate gradients (left) and minimum residuals (right).*

In Figure 6.2 we show the effect of our preconditioning approach on the spectrum of the matrix. We have used the same mesh Reynolds numbers, but with a 32×32

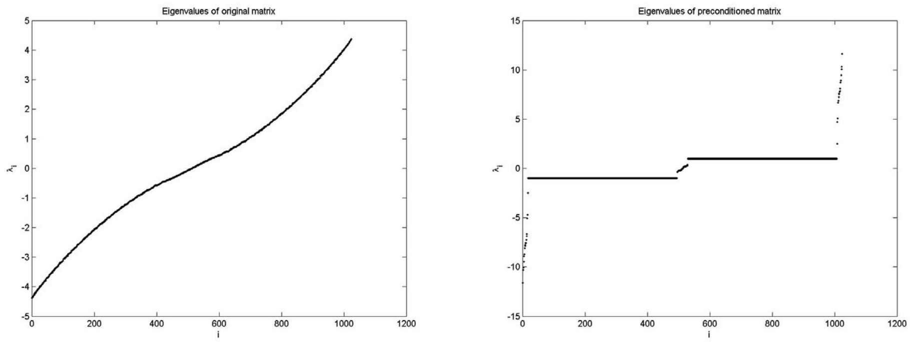

FIG. 6.2. *Imaginary part of the eigenvalues of the original matrix (left) and the preconditioned matrix (right). The two large clusters of eigenvalues on the right-hand plot correspond to the eigenvalues $\imath$ and $-\imath$.*

mesh; the matrix is 1024 in dimension. We see that preconditioning generates very strong clustering near $\pm\imath$.

7. Conclusions. We have considered conjugate gradients and minimum residual algorithms for the skew-symmetric system. Applying Galerkin conditions, the CG scheme requires two matrix-vector products per iteration and is equivalent to CGNE, which works on the normal equations, but it is possible to identify the conjugacy of the search directions in terms of the original skew-symmetric matrix. A stable 2×2 block decomposition of the tridiagonal matrix that arises from a skew-symmetric version of the Lanczos algorithm is central in the derivation.

Preconditioning is crucial for convergence, and we have presented block incomplete factorizations that maintain the skew-symmetry of the (preconditioned) matrix.

As expected, Krylov subspace iterative solvers perform like solvers for symmetric indefinite systems with symmetric spectrum about the origin. But the special structure of the skew-symmetric matrix gives rise to a unique derivation of the algorithms.

Future work may include an extensive set of numerical experiments and an investigation of the effectiveness of incorporating skew-symmetric solvers into the numerical solution of general linear systems with a dominant skew-symmetric part.

REFERENCES

[1] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] P. BENNER, R. BYERS, H. FASSBENDER, V. MEHRMANN, AND D. WATKINS, *Cholesky-like factorizations of skew-symmetric matrices*, Electron. Trans. Numer. Anal., 11 (2000), pp. 85–93 (electronic).

[3] J. R. BUNCH, *A note on the stable decomposition of skew-symmetric matrices*, Math. Comp., 38 (1982), pp. 475–479.

[4] X.-W. CHANG, *Some features of Gaussian elimination with rook pivoting*, BIT, 42 (2002), pp. 66–83.

[5] P. CONCUS AND G. H. GOLUB, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Sciences and Engineering (Versailles, 1975), Part 1, Lecture Notes in Econom. and Math. Systems 134, Springer, Berlin, 1976, pp. 56–65.

[6] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[7] I. S. DUFF, *The Design and Use of a Sparse Direct Solver for Skew Symmetric Matrices*, Technical report TR/PA/07/04, CERFACS, Toulouse, France, 2007.

[8] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.

[9] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.

[10] C. Gu and H. Qian, *Skew-symmetric methods for nonsymmetric linear systems with multiple right-hand sides*, J. Comput. Appl. Math., 23 (2009), pp. 567–577.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[12] Y. Huang, A. J. Wathen, and L. Li, *An iterative method for skew-symmetric systems*, Information, 2 (1999), pp. 147–153.

[13] R. Idema and C. Vuik, *A minimal residual method for shifted skew-symmetric systems*, Tech. report 07-09, Delft University of Technology, Delft, The Netherlands, 2007.

[14] E. Jiang, *Algorithm for solving shifted skew-symmetric linear system*, Front. Math. China, 2 (2007), pp. 227–242.

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