Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Volker Mehrmann David Watkins

Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils

Preprint SFB393/00-02

Preprint-Reihe des Chemnitzer SFB 393

SFB393/00-02 **January 2000**

Contents

Authors:

Volker Mehrmann
Fakultät für Mathematik
Technische Universität Chemnitz
D-09107 Chemnitz, FRG
`mehrmann@mathematik.tu-chemnitz.de`

David Watkins
Washington State Univerity
Pullman, WA, 99164-3113, USA
currently: Fakultät für Mathematik
Technische Universität Chemnitz,
D-09107 Chemnitz, FRG

Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils

Volker Mehrmann[1] David Watkins[2]

Abstract

We study large, sparse generalized eigenvalue problems for matrix pencils, where one of the matrices is Hamiltonian and the other skew Hamiltonian. Problems of this form arise in the numerical simulation of elastic deformation of anisotropic materials, in structural mechanics and in the linear-quadratic control problem for partial differential equations. We develop a structure-preserving skew-Hamiltonian, isotropic, implicitly-restarted shift-and-invert Arnoldi algorithm (SHIRA). Several numerical examples demonstrate the superiority of SHIRA over a competing unstructured method.

Keywords: Skew-Hamiltonian/Hamiltonian pencil, generalized eigenvalue problem, quadratic eigenvalue problem, implicitly restarted Arnoldi method, Lamé equations, classical mechanics, linear quadratic control, algebraic Riccati equation

1 Introduction

In this paper we study the numerical computation of a small number of eigenvalues (and the associated eigenvectors) of large-scale generalized eigenvalue problems having a certain structure that arises frequently in applications. A $2n \times 2n$ real matrix pencil

$$\alpha\mathcal{N} - \beta\mathcal{H} = \alpha \begin{bmatrix} F_1 & G_1 \\ H_1 & F_1^T \end{bmatrix} - \beta \begin{bmatrix} F_2 & G_2 \\ H_2 & -F_2^T \end{bmatrix}, \tag{1}$$

where $G_1 = -G_1^T$, $H_1 = -H_1^T$, $G_2 = G_2^T$, and $H_2 = H_2^T$, is called a *skew-Hamiltonian/ Hamiltonian* pencil or, more briefly, an *SHH* pencil. Throughout this paper we will assume that $\mathcal{N}$ and $\mathcal{H}$ are large and sparse. Several examples of applications that give rise to large, sparse generalized eigenvalue problems with SHH pencils are given in Section 2.

The reason for the terminology is simply this: Matrices the form of $\mathcal{N}$ and $\mathcal{H}$ in (1) are called *skew Hamiltonian* and *Hamiltonian*, respectively. If

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}, \tag{2}$$

where I_n is the $n \times n$ identity matrix, then skew-Hamiltonian matrices satisfy $(\mathcal{N}J)^T = -(\mathcal{N}J)$ and Hamiltonian matrices satisfy $(\mathcal{H}J)^T = \mathcal{H}J$.

[1] This work has been supported by *Deutsche Forschungsgemeinschaft within SFB393 Numerische Simulation auf massiv parallelen Rechnern.*

[2] This work has been supported by *Deutsche Forschungsgemeinschaft within project A8 of SFB393 Numerische Simulation auf massiv parallelen Rechnern* and by Deutscher Akademischer Austauschdienst within the program HSP III, Förderung ausländischer Gastdozenten.

An SHH pencil and in particular its spectrum have considerable structure. The eigenvalues occur in quadruplets $(\lambda, \bar{\lambda}, -\lambda, -\bar{\lambda})$, or in real or purely imaginary pairs $(\lambda, -\lambda)$ [23, 24, 25]. The objective of this paper is to develop algorithms that preserve and exploit this structure. The payoffs are more efficient and more accurate algorithms. In some cases preservation of the structure is crucial to the stable solution of a problem.

Typical applications require the few eigenvalues that are smallest in magnitude or closest to the imaginary axis. To achieve this we must apply transformations that have the effect of shifting the desired eigenvalues to the periphery of the spectrum, so that they can be computed efficiently by Krylov subspace methods, e.g. Arnoldi or Lanczos, possibly with implicit restarts. This is a standard procedure in methods for large sparse eigenvalue problems, [30, 37]. What is special to this paper is that our transformations and our Krylov subspace methods respect the structure of the problem.

Our approach requires that the skew-Hamiltonian matrix $\mathcal{N}$ be presented as a product in the special form

$$\mathcal{N} = \mathcal{Z}_1 \mathcal{Z}_2 \quad \text{where} \quad \mathcal{Z}_2^T J = \pm J \mathcal{Z}_1, \tag{3}$$

with $\mathcal{Z}_1$ and $\mathcal{Z}_2$ sparse. This allows us to transform the pencil $\alpha\mathcal{N} - \beta\mathcal{H}$ to a standard eigenvalue problem $\alpha I - \beta\mathcal{W} = \alpha I - \beta\mathcal{Z}_1^{-1}\mathcal{H}\mathcal{Z}_2^{-1}$, in which the matrix $\mathcal{W} = \mathcal{Z}_1^{-1}\mathcal{H}\mathcal{Z}_2^{-1}$ turns out to be Hamiltonian.

This procedure is analogous to the technique by which a symmetric pencil $A - \lambda B$, with B positive definite, is transformed to a standard eigenvalue problem using a Cholesky decomposition $B = LL^T$. Because of the symmetry of the decomposition, the matrix $L^{-1}AL^{-T}$ inherits the symmetry of A. In the current context, if we introduce the skew-symmetric "inner product" $\langle x, y\rangle_J = y^T Jx$, we find that a matrix is Hamiltonian if and only if it is skew symmetric with respect to this J-inner product, i.e. $\langle Hx, y\rangle_J = -\langle x, Hy\rangle_J$ for all $x, y \in \mathbf{R}^{2n}$. The relationship between $\mathcal{Z}_1$ and $\mathcal{Z}_2$ given in (3) implies that $\mathcal{Z}_2$ is (plus or minus) the adjoint of $\mathcal{Z}_1$ with respect to the J-inner product, i.e. $\langle \mathcal{Z}_1 x, y\rangle_J = \pm\langle x, \mathcal{Z}_2 y\rangle_J$ for all $x, y \in \mathbf{R}^{2n}$. Thus the decomposition $\mathcal{N} = \mathcal{Z}_1\mathcal{Z}_2$ is a symmetric, Cholesky-like decomposition of $\mathcal{N}$. Consequently $\mathcal{W} = \mathcal{Z}_1^{-1}\mathcal{H}\mathcal{Z}_2^{-1}$ inherits the J-skew symmetry of $\mathcal{H}$; that is, $\mathcal{W}$ is Hamiltonian.

We discuss two approaches that make different transformations of the Hamiltonian operator $\mathcal{W}$. The first maps $\mathcal{W}$ to a skew-Hamiltonian operator, from which the eigenvalues can be extracted by an implicitly restarted Arnoldi method that has been modified to preserve the structure. For this approach we provide numerical results demonstrating its effectiveness. The second approach maps $\mathcal{W}$ to a symplectic operator by a generalized Cayley transform. The desired eigenvalues can then be extracted by an implicitly restarted symplectic Lanczos method. We only outline this approach and discuss its advantages and disadvantages.

2 Applications

The need to solve SHH generalized eigenvalue problems arises in many applications. The best-known example is the linear quadratic optimal control problem for descriptor systems, where the pencil typically has the particular form

$$\alpha \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix} - \beta \begin{bmatrix} A & -BB^T \\ C^TC & -A^T \end{bmatrix}, \tag{4}$$

with B of size $n \times m$, C of size $p \times n$ and $m << n, p << n$, see [3, 4, 25]. Large sparse problems of this type arise for example in the control of semidiscretized parabolic partial differential equations [17, 28, 29].

Here the skew-Hamiltonian matrix can be written in factored form as

$$\mathcal{N} = \mathcal{Z}_1\mathcal{Z}_2 = \begin{bmatrix} E & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & E^T \end{bmatrix}, \tag{5}$$

with $\mathcal{Z}_2^T J = J\mathcal{Z}_1$.

The application that we will discuss in detail in this paper arises from quadratic eigenvalue problems of the form

$$\lambda^2 Mx + \lambda Gx + Kx = 0, \tag{6}$$

where $M = M^T$ is positive definite, $K = K^T$ and $G = -G^T$.

Large sparse eigenvalue problems of this form arise for example in finite element discretization in structural analysis [33], in acoustic simulation of poro-elastic materials [22, 31, 34], and in the elastic deformation of anisotropic materials [18, 20, 32]. In these applications M is a mass matrix and $-K$ a stiffness matrix. Depending on the applications, different parts of the spectrum are of interest, typically one is interested in the eigenvalues with smallest real part or the eigenvalues smallest or largest in modulus.

At first glance the quadratic eigenvalue problem (6) and the SHH generalized eigenvalue problem (1) seem not to have much in common. However, it is well known that the eigenvalues of (6) occur in quadruplets $(\lambda, \bar{\lambda}, -\lambda, -\bar{\lambda})$ or real or purely imaginary pairs $(\lambda, -\lambda)$ [19], just as they do for (1). The reason for this similarity is that (6) can be 'linearized' to have the form (1). This can be done in several different ways. For example, if we make the substitution

$$y = \lambda Mx \tag{7}$$

in (6) and rewrite the substitution as $\lambda x - M^{-1}y = 0$, we obtain the SHH generalized eigenvalue problem

$$\lambda\mathcal{N}z - \mathcal{H}z = \lambda \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} 0 & -K \\ M^{-1} & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0. \tag{8}$$

Since $\mathcal{N}$ is invertible, the pencil $\lambda\mathcal{N} - \mathcal{H}$ is regular, i.e., $\det(\lambda\mathcal{N} - \mathcal{H})$ does not vanish identically.

The skew Hamiltonian matrix $\mathcal{N}$ can be written in factored form as

$$\mathcal{N} = \mathcal{Z}_1\mathcal{Z}_2 = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix}, \tag{9}$$

with $\mathcal{Z}_2^T J = J\mathcal{Z}_1$. We note that $\mathcal{Z}_1 = \mathcal{Z}_2$ is a skew-Hamiltonian square root of $\mathcal{N}$.

A second approach is to use the substitution

$$y = \lambda x \tag{10}$$

in (6) and rewrite the substitution as $\lambda M x - M y = 0$. This yields the SHH generalized eigenvalue problem

$$\lambda \mathcal{N} z - \mathcal{H} z = \lambda \begin{bmatrix} M & G \\ 0 & M \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} 0 & -K \\ M & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0. \tag{11}$$

In this case the skew Hamiltonian matrix $\mathcal{N}$ can be written in factored form as

$$\mathcal{N} = \mathcal{Z}_1 \mathcal{Z}_2 = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix} \begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix}, \tag{12}$$

with $\mathcal{Z}_2^T J = J \mathcal{Z}_1$.

An approach that is intermediate to the previous two uses the Cholesky factorization $M = LL^T$ and the substitution

$$y = \lambda L^T x \tag{13}$$

in (6) to obtain the SHH generalized eigenvalue problem

$$\lambda \mathcal{N} z - \mathcal{H} z = \lambda \begin{bmatrix} L & G \\ 0 & L^T \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} 0 & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0. \tag{14}$$

The skew Hamiltonian matrix $\mathcal{N}$ can be written in the factored form

$$\mathcal{N} = \mathcal{Z}_1 \mathcal{Z}_2 = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & L^T \end{bmatrix} \begin{bmatrix} L & \frac{1}{2}G \\ 0 & I \end{bmatrix}, \tag{15}$$

with $\mathcal{Z}_2^T J = J \mathcal{Z}_1$.

It turns out that all three of these linearizations give rise to the same Hamiltonian operator $\mathcal{W} = \mathcal{Z}_1^{-1} \mathcal{H} \mathcal{Z}_2^{-1}$, so which one we use is just a matter of convenience.

If the matrix K is nonsingular, the substitution (10) together with the equivalent equation $Ky = \lambda K x$ leads to the Hamiltonian/skew-Hamiltonian generalized eigenvalue problem

$$\lambda \begin{bmatrix} 0 & -M \\ K & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} K & G \\ 0 & K \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0. \tag{16}$$

We prefer not to use this linearization. If K is nearly singular, as is typically the case in practice, then the matrix pencil in equation (16) is close to a singular pencil [11].

Other linearizations that have been used are

$$\lambda \begin{bmatrix} I & 0 \\ G & M \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & I \\ -K & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0. \tag{17}$$

or equivalently

$$\lambda \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & I \\ -K & -G \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0, \tag{18}$$

which are both obtained with the substitution (10). Neither (17) nor (18) is SHH. Although in particular (18) seems attractive, since the matrix $\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}$ is Hermitian positive definite, it is still preferable to use an SHH linearization, which reflects the structure of the spectrum. Since we construct methods that respect this structure also in finite arithmetic, the computed eigenvalues will be the exact eigenvalues of a perturbed SHH pencil. Thus the computed eigenvalues will occur in quadruplets, as they should. See the perturbation analysis in [3, 8, 9, 10]. Furthermore, the comparison of the perturbation theory for quadratic eigenvalue problems and the related linearization, as it has been analyzed in [39], shows that even numerically backwards stable methods that do not respect the special structure of the quadratic eigenvalue problem in its linearization may be unstable.

Surveys of numerical methods for the quadratic eigenvalue problem (in various forms) have been given in [22, 35, 36]. In particular Lanzcos and Arnoldi methods as well as Jacobi-Davidson iterations are compared. Simultaneous iteration methods for quadratic eigenvalue problems have been studied in [13, 27, 36].

Some of the methods can be directly applied to the quadratic problem while others only work for the linearizations. The major difference between the direct solution of the quadratic problem and its linearizations is that in order to solve the projected problems and to use implicit restarts one needs some kind of generalized Schur form for the problem. Such a form is not known for quadratic problems, while for matrix pencils this is the generalized Schur form [16], and for SHH pencils there is a structured generalized form that was analyzed in [23, 24]. Numerical methods for its computation have been derived in [3, 4].

3 Rational transformations of Hamiltonian matrices

If the skew-Hamiltonian matrix $\mathcal{N}$ in the SHH pencil $\alpha\mathcal{N} - \beta\mathcal{H}$ is invertible and given in the factored form $\mathcal{N} = \mathcal{Z}_1\mathcal{Z}_2$ with $\mathcal{Z}_2^T J = \pm J\mathcal{Z}_1$, then the pencil is equivalent to the pencil

$$\alpha I - \beta\mathcal{W} := \alpha I - \beta\mathcal{Z}_1^{-1}\mathcal{H}\mathcal{Z}_2^{-1}. \tag{19}$$

As we explained in the introduction, $\mathcal{W}$ is again Hamiltonian. Hence we can consider first transformations of Hamiltonian matrices, then apply these to SHH pencils by rewriting them in terms of the original data.

Suppose we want to find the eigenvalues of $\mathcal{W}$ that lie nearest to some target value λ_0. The standard transformation for this purpose is $(\mathcal{W} - \lambda_0 I)^{-1}$, which however fails to preserve the structure. Each eigenvalue near λ_0 is related to eigenvalues near $\bar{\lambda}_0$, $-\lambda_0$, and $-\bar{\lambda}_0$. If we are to hope to preserve the structure, we must extract all four of these eigenvalues together. Thus we should also bring $(\mathcal{W}-\bar{\lambda}_0 I)^{-1}$, $(\mathcal{W}+\lambda_0 I)^{-1}$, and $(\mathcal{W}+\bar{\lambda}_0 I)^{-1}$ into play.

3.1 Transformation to a skew-Hamiltonian operator

For finding the eigenvalues nearest the quadruplet $(\lambda_0, \bar{\lambda}_0, -\lambda_0, -\bar{\lambda}_0)$ the obvious rational transformation is

$$R_1(\lambda_0, \mathcal{W}) = (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I)^{-1}(\mathcal{W} - \bar{\lambda}_0 I)^{-1}(\mathcal{W} + \bar{\lambda}_0 I)^{-1}. \tag{20}$$

If the target λ_0 is either real or purely imaginary, one may choose to use the simpler transformation

$$R_2(\lambda_0, \mathcal{W}) = (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I)^{-1}. \tag{21}$$

Both R_1 and R_2 turn out to be skew Hamiltonian. In what follows we will focus on the operator R_1. In every case we can make similar assertions about R_2.

The transformation $R_1(\lambda_0, \mathcal{W})$ maps all eigenvalues near to $\lambda_0, -\lambda_0, \bar{\lambda}_0, -\bar{\lambda}_0$ simultaneously to values of large modulus. Hence fast convergence of the eigenvalues near these points can be expected if we apply any of the standard iterative methods, [30, 37].

In order to apply an iterative method such as subspace iteration or the Arnoldi process, we need to be able multiply the matrix R_1 by an arbitrary vector at reasonable cost, since this operation is performed repeatedly by these algorithms. Thus we need to be able to apply operators of the form $(\mathcal{W} - \sigma I)^{-1}$ inexpensively. We can either work with $\mathcal{W}$ directly or we can refer back to the original data $\mathcal{H}$ and $\mathcal{N}$. Substituting $\mathcal{Z}_1^{-1}\mathcal{H}\mathcal{Z}_2^{-1}$ for $\mathcal{W}$ in (20) and simplifying, we find that

$$\begin{aligned} R_1 &= R_1(\lambda_0, \mathcal{H}, \mathcal{N}, \mathcal{Z}_1, \mathcal{Z}_2) \qquad (22) \\ &= \mathcal{Z}_2(\mathcal{H} - \lambda_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} + \lambda_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} - \bar{\lambda}_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} + \bar{\lambda}_0\mathcal{N})^{-1}\mathcal{Z}_1. \end{aligned}$$

Since $\mathcal{Z}_1$, $\mathcal{Z}_2$, and $\mathcal{N}$ are sparse matrices, they can be applied easily. The only question, then, is how to apply operators of the form $(\mathcal{H} - \sigma\mathcal{N})^{-1}$ inexpensively. This question will be discussed in connection with specific applications in Section 4.

We now consider the structural properties of $R_1(\lambda_0, \mathcal{W})$ and $R_2(\lambda_0, \mathcal{W})$, beginning with a well known lemma.

Lemma 1
(i) If $\mathcal{W}$ is Hamiltonian, then $\mathcal{W}^2$ is skew Hamiltonian.
(ii) If $\mathcal{W}$ is skew Hamiltonian, then $\mathcal{W}^2$ is skew Hamiltonian.
(iii) If $\mathcal{W}$ is skew Hamiltonian and invertible, then $\mathcal{W}^{-1}$ is skew Hamiltonian.

We omit the proof, which is straightforward, see e.g. [1].

Proposition 2 *If $\mathcal{W}$ is a real Hamiltonian matrix, then $R_1(\lambda_0, \mathcal{W})$ in (20) is real skew Hamiltonian. If, in addition, λ_0 is either real or purely imaginary, then $R_2(\lambda_0, \mathcal{W})$ in (21) is also real and skew Hamiltonian.*

Proof. Direct calculation shows that

$$R_1(\lambda_0, \mathcal{W}) = (|\lambda_0|^4 I + [2|\lambda_0|^2 - 4(\Re(\lambda_0))^2]\mathcal{W}^2 + \mathcal{W}^4)^{-1}.$$

Hence $R_1(\lambda_0, \mathcal{W})$ is a real matrix, and by Lemma 1 it is skew Hamiltonian, since the skew Hamiltonian matrices form a real vector space that contains the identity matrix.

A similar but simpler argument shows that $R_2(\lambda_0, \mathcal{W})$ is skew Hamiltonian. □

Notice that the eigenvalues λ and $-\lambda$ of $\mathcal{W}$ both get mapped to $\mu = (\lambda^4 + c_2\lambda^2 + c_0)^{-1}$ of R_1 and to $\nu = (\lambda^2 - \lambda_0^2)^{-1}$ of R_2. Thus all eigenvalues of R_1 and R_2 have even multiplicity. This fact, which is true of skew Hamiltonian matrices in general, must be taken into account by our numerical methods.

Recall that the *Krylov subspace* $\mathcal{K}_j(A, v)$ is defined by

$$\mathcal{K}_j(A, v) = \text{span}\{v, Av, A^2v, \ldots, A^{j-1}v\}.$$

The Arnoldi method and other Krylov subspace methods generate Krylov subspaces, where v is the starting vector and A is the operator being applied, e.g. R_1 in our case, see [30]. Recall, furthermore, that a subspace $\mathcal{S}$ of $\mathbf{R}^{2n}$ is called *isotropic* iff $y^TJx = 0$ for all x, $y \in \mathcal{S}$. In other words, $\mathcal{S}$ is isotropic iff $J\mathcal{S}$ is orthogonal to $\mathcal{S}$ with respect to the standard inner product on $\mathbf{R}^{2n}$, see [2]. The beauty of skew-Hamiltonian operators is that the Krylov subspaces that they generate are isotropic.

Proposition 3 *Let $A \in \mathbf{R}^{2n\times 2n}$ be a skew-Hamiltonian matrix, let $v \in \mathbf{R}^{2n}$, and let j be a positive integer. Then the Krylov subspace $\mathcal{K}_j(A, v)$ is isotropic.*

Proof. Since A is real skew Hamiltonian, one easily shows that all of its powers A^i, $i = 0, 1, 2, 3, \ldots$, are also skew Hamiltonian. This means that JA^i is skew symmetric. To establish isotropy of $\mathcal{K}_j(A, v)$, it suffices, by linearity, to prove that $(A^iv)^TJA^kv = 0$ for all $i \geq 0$ and $k \geq 0$. Since $A^TJ = JA$, we have $(A^iv)^TJA^kv = v^TJA^{i+k}v$, which equals zero because JA^{i+k} is skew symmetric. □

We now introduce an *isotropic* Arnoldi process, which will form the basis for our numerical method. At first we allow A to be an arbitrary $2n \times 2n$ real matrix. Recall that the Arnoldi process starts with an arbitrary unit vector q_1 and produces orthonormal vectors as follows. Given orthonormal vectors $q_1, \ldots, q_j$, the next vector q_{j+1} is generated by forming Aq_j and then orthogonalizing it against $q_1, \ldots, q_j$. Thus

$$q_{j+1}h_{j+1,j} = Aq_j - \sum_{i=1}^{j} q_ih_{ij},$$

where $h_{ij} = q_i^TAq_j$, $i = 1, \ldots, j$, and $h_{j+1,j}$ is a positive constant chosen so that $\|q_{j+1}\|_2 = 1$.

If we wish to build isotropic subspaces, we should orthogonalize against $Jq_1, \ldots, Jq_j$ as well. Thus the j-th step of the *isotropic Arnoldi process* is

$$q_{j+1}h_{j+1,j} = Aq_j - \sum_{i=1}^{j} q_ih_{ij} - \sum_{i=1}^{j} Jq_it_{ij}, \tag{23}$$

where

$$h_{ij} = q_i^TAq_j \quad \text{and} \quad t_{ij} = (Jq_i)^TAq_j, \tag{24}$$

and $h_{j+1,j}$ is a positive constant chosen so that $\|q_{j+1}\|_2 = 1$. This generates orthonormal vectors that span isotropic subspaces.

If A is real skew Hamiltonian, then the coefficients t_{ij} in (23,24) will all be zero, by Proposition 3. Thus the isotropic Arnoldi process reduces to the ordinary Arnoldi process in this case, at least in theory.

If the quantity on the right-hand side of (23) is zero, then the process breaks down. It can be continued by setting $h_{j+1,j} = 0$ and taking q_{j+1} to be any unit vector that is orthogonal to $q_1, \ldots, q_j$ and $Jq_1, \ldots, Jq_j$. Equation (23) holds in this case, too. In exact arithmetic, the process terminates after $n-1$ steps, as $q_1, \ldots, q_n$ together with $Jq_1, \ldots, Jq_n$ form an orthonormal basis of $\mathbf{R}^{2n}$.

Let $Q \in \mathbf{R}^{2n\times n}$ be the matrix whose columns are $q_1, \ldots, q_n$. Then (23) can be rewritten as

$$AQ = \left[\begin{array}{cc} Q & JQ \end{array}\right] \left[\begin{array}{c} H \\ T \end{array}\right], \tag{25}$$

where H is an upper Hessenberg matrix built from the coefficients h_{ij}, and T is an upper triangular matrix built from the coefficients t_{ij}.

Recall that a matrix $S \in \mathbf{R}^{2n\times 2n}$ is *symplectic* if $S^T JS = J$. The next lemma recalls a few simple properties that we need, see [2].

Lemma 4

(i) If the columns of $Q \in \mathbf{R}^{2n\times n}$ are orthonormal and span an isotropic subspace, then the matrix $\left[\begin{array}{cc} Q & JQ \end{array}\right]$ is both orthogonal and symplectic.

(ii) If $\mathcal{N}$ is skew Hamiltonian and S is symplectic, then $S^{-1}\mathcal{N}S$ is skew Hamiltonian.

Equation (25) implies that

$$A\left[\begin{array}{cc} Q & JQ \end{array}\right] = \left[\begin{array}{cc} Q & JQ \end{array}\right] \left[\begin{array}{cc} H & N \\ T & P \end{array}\right] \tag{26}$$

for some N and P. The matrix $\mathcal{U} = \left[\begin{array}{cc} Q & JQ \end{array}\right]$ is orthogonal and symplectic by Lemma 4, and we have

$$\mathcal{U}^{-1}A\mathcal{U} = \left[\begin{array}{cc} H & N \\ T & P \end{array}\right].$$

If A is skew Hamiltonian, then, since the skew Hamiltonian property is preserved under symplectic similarity by Lemma 4, we must have $P = H^T$, $T^T = -T$, and $N^T = -N$. Since T is upper triangular, the additional equation $T^T = -T$ implies $T = 0$, which we had already noted. This gives the following result, which is not new [40].

Proposition 5 *Let $\mathcal{N}$ be a real skew Hamiltonian matrix. Then for every unit vector $q_1 \in \mathcal{R}^{2n}$ there exists an orthogonal symplectic matrix $\mathcal{U}$ that has q_1 as its first column such that*

$$\mathcal{U}^T\mathcal{N}\mathcal{U} = \left[\begin{array}{cc} H & N \\ 0 & H^T \end{array}\right], \tag{27}$$

and H is in Hessenberg form.

The multiplicity of the eigenvalues of $\mathcal{N}$ is reflected in the structure of the matrix in (27). For each double eigenvalue of $\mathcal{N}$, one copy resides in H, and the other copy is in H^T.

When we apply the Arnoldi process to a large, sparse matrix in practice, we have neither the time nor the storage space to carry the process to completion. Instead we stop after k steps with $k \ll n$. The coefficients h_{ij} computed to this point, form a $k \times k$ submatrix of H whose eigenvalues (*Ritz values*) we can compute and use as estimates of eigenvalues of H, hence of $\mathcal{N}$. Since the eigenvalues do not appear in duplicate in H, we get each eigenvalue once, not twice. This is important to the success of the numerical method. If we make the effort to compute, say, six eigenvalues, we would like to get six distinct eigenvalues, not three eigenvalues in duplicate.

The developments outlined here are valid only for real matrices. However, it has been shown in [7] that for complex problems an embedding of the problem into a double sized real problem can be used to get a real skew Hamiltonian problem from which all the spectral information can be obtained.

3.2 Transformation to symplectic form

Another natural rational transformation that is sometimes applied to Hamiltonian matrices is the generalized Cayley transform. See [26] for a detailed analysis. Given a target λ_0, the generalized Cayley transform of the Hamiltonian matrix $\mathcal{W}$ is

$$S_1(\lambda_0, \mathcal{W}) = (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \bar{\lambda}_0 I)(\mathcal{W} - \bar{\lambda}_0 I)^{-1}(\mathcal{W} + \lambda_0 I). \tag{28}$$

By using four factors we obtain a real matrix, even when the target λ_0 is complex. If λ_0 is real, we can use the simpler two-factor Cayley transform

$$S_2(\lambda_0, \mathcal{W}) = (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I). \tag{29}$$

Notice that for $i = 1,\ 2$, $S_i^{-1}(\lambda) = S_i(-\lambda)$. Thus it is no more expensive to apply S_i^{-1} than S_i.

In terms of the original data we have

$$\begin{aligned} S_1 &= S_1(\lambda_0, \mathcal{H}, \mathcal{N}, \mathcal{Z}_1, \mathcal{Z}_2) \\ &= \mathcal{Z}_2(\mathcal{H} - \lambda_0\mathcal{N})^{-1}(\mathcal{H} + \bar{\lambda}_0\mathcal{N})(\mathcal{H} - \bar{\lambda}_0\mathcal{N})^{-1}(\mathcal{H} + \lambda_0\mathcal{N})\mathcal{Z}_2^{-1}. \end{aligned} \tag{30}$$

A similar expression holds for S_2.

We now discuss the structural properties of S_1 and S_2. The following proposition is well known [26].

Proposition 6

(i) If $\mathcal{W} \in \mathbf{R}^{2n\times 2n}$ is Hamiltonian, then $S_1(\lambda_0, \mathcal{W})$ in (28) is real symplectic.

(ii) If $\mathcal{W} \in \mathbf{R}^{2n\times 2n}$ is Hamiltonian, and λ_0 is real, then $S_2(\lambda_0, \mathcal{W})$ in (29) is real symplectic.

The use of symplectic matrices has several advantages and disadvantages compared with skew-Hamiltonian matrices. First of all, in contrast to the skew Hamiltonian case, not every symplectic matrix has a symplectic Schur form [21]. What is worse in the symplectic case is that a symplectic Hessenberg like form only exists for very special first columns of the transformation matrix [2]. Thus Arnoldi methods, with or without restarts, cannot be used with symplectic matrices, which is definitely a disadvantage of this rational transformation.

The transformation S_1 maps all eigenvalues near $\lambda_0, \bar{\lambda}_0$ simultaneously to values of large modulus. At the same time the eigenvalues near $-\lambda_0, -\bar{\lambda}_0$ are mapped close to 0. These correspond to large eigenvalues of the inverse transformation S_1^{-1}. The coexistence of extremely large and small eigenvalues is an inevitable consequence of the fact that eigenvalues of a symplectic matrix occur in quadruplets $\mu, \bar{\mu}, \mu^{-1}, \bar{\mu}^{-1}$. Any method that preserves the symplectic structure must extract quadruplets intact. This can be achieved by structure-preserving Lanczos-like methods that employ both S and S^{-1} in a symmetric manner. See [5, 6, 15] for structure-preserving implicitly-restarted Lanczos-like methods applicable to symplectic problems. These are the sorts of methods we must apply to S_1 and S_2.

Another disadvantage of the Cayley transform is that it is effective only if the target λ_0 is not too close to the imaginary axis. Notice that if λ_0 is purely imaginary, then $S_1 = I$, which is not useful for extracting spectral information about $\mathcal{W}$.

An advantage of the Cayley transform approach is that, unlike the skew-Hamiltonian approach, it can be extended to complex matrices and matrix pencils in a straightforward way.

4 Applying the operators

In our two applications the matrices $\mathcal{N}$, $\mathcal{H}$, $\mathcal{Z}_1$, and $\mathcal{Z}_2$ have further structure that can be used to simplify the formulas.

4.1 Quadratic eigenproblems

Let us first study the quadratic eigenvalue problem in the linearization (11) with $\mathcal{N}$ in factored form (12). First of all

$$\begin{aligned} \mathcal{H} - \sigma\mathcal{N} &= -\begin{bmatrix} \sigma M & \sigma G + K \\ -M & \sigma M \end{bmatrix} \\ &= \begin{bmatrix} I & -\sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -Q(\sigma) \\ M & 0 \end{bmatrix} \begin{bmatrix} I & -\sigma I \\ 0 & I \end{bmatrix}, \end{aligned} \tag{31}$$

where

$$Q(\sigma) := \sigma^2 M + \sigma G + K. \tag{32}$$

Thus

$$(\mathcal{W} - \sigma I)^{-1} = \mathcal{Z}_2(\mathcal{H} - \sigma\mathcal{N})^{-1}\mathcal{Z}_1 =$$

$$\begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & M^{-1} \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & \sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}. \tag{33}$$

We can obtain an expression that does not involve M^{-1} by using one or the other of the factorizations

$$\begin{bmatrix} 0 & M^{-1} \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & M^{-1} \end{bmatrix} \tag{34}$$

and

$$\begin{bmatrix} 0 & M^{-1} \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} = \begin{bmatrix} M^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} \tag{35}$$

Substituting (34) into (33) and combining the last three matrices, we obtain

$$(\mathcal{W} - \sigma I)^{-1} = \begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G + \sigma M \\ 0 & I \end{bmatrix}. \tag{36}$$

If we use (35) instead of (34), we obtain instead

$$(\mathcal{W} - \sigma I)^{-1} = \begin{bmatrix} I & \frac{1}{2}G + \sigma M \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\sigma)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & \sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}. \tag{37}$$

Both of these expressions are useful.

To apply this operator in either form (36) or (37) we will need a sparse LU decomposition of the sparse, nonsymmetric matrix $Q(\sigma)$. In $R_1(\lambda_0, \mathcal{W})$ there are four factors of the form $(\mathcal{W} - \sigma I)^{-1}$, corresponding to $\sigma = \pm\lambda_0$ and $\sigma = \pm\bar{\lambda}_0$. Thus it might appear that we need four sparse LU factorizations in order to apply R_1. However, one immediately observes that

$$\begin{aligned} Q(\bar{\sigma}) &= \bar{\sigma}^2 M + \bar{\sigma} G + K = \overline{Q(\sigma)}, \\ Q(-\sigma) &= \sigma^2 M - \sigma G + K = Q(\sigma)^T, \\ Q(-\bar{\sigma}) &= \bar{\sigma}^2 M - \bar{\sigma} G + K = \overline{Q(\sigma)}^T. \end{aligned} \tag{38}$$

An LU factorization of $Q(\sigma)$ immediately yields factorizations of $Q(\bar{\sigma})$, $Q(-\sigma)$, and $Q(-\bar{\sigma})$. Thus one sparse LU factorization is all we need.

To derive the formulas for applying the operator R_1, we first discuss the application of R_2. If we combine expressions (36) and (37), replacing σ by λ_0 in (36) and by $-\lambda_0$ in (37), we obtain

$$\begin{aligned} R_2(\lambda_0, \mathcal{W}) &= (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I)^{-1} \\ &= \begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \lambda_0 I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\lambda_0)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \\ &\quad \times \begin{bmatrix} 0 & I \\ -Q(-\lambda_0)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & -\lambda_0 I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}, \end{aligned} \tag{39}$$

which can be readily translated into an algorithm for applying the operator R_2, i.e. for multiplying R_2 by a vector. For example, to multiply the factor $\begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}$ by the vector $\begin{bmatrix} x \\ y \end{bmatrix}$, we only need to perform the operations $x \leftarrow x + \frac{1}{2}Gy$ and $y \leftarrow My$, which requires two sparse $n \times n$ matrix-vector products and one saxpy operation. To multiply $\begin{bmatrix} 0 & I \\ -Q(-\lambda_0)^{-1} & 0 \end{bmatrix}$ by $\begin{bmatrix} x \\ y \end{bmatrix}$, we perform the operations $\hat{x} \leftarrow -x$, $x \leftarrow y$, and $y \leftarrow Q(\lambda_0)^{-1}\hat{x}$. Given a sparse LU factorization of $Q(\lambda_0)$, we obtain $y \leftarrow Q(\lambda_0)^{-1}\hat{x}$ by two sparse triangular solves. Applying these observations to (39) we easily determine the total work for multiplying $R_2(\lambda_0, \mathcal{W})$ by a vector.

Table 1 Operation count for applying the operator R_2, assuming a sparse LU factorization of $Q(\lambda_0)$ is available.

4	sparse triangular solves
2	symmetric sparse matrix-vector products Mz
3	skew symmetric sparse matrix-vector products Gz
5	saxpy operations.

The sparse LU factorization of $Q(\lambda_0)$ needs to be computed only once; then R_2 can be applied repeatedly. No further LU factorizations are needed unless the target λ_0 is changed. If λ_0 is real, then all operations are real.

Since $R_1(\lambda_0, \mathcal{W}) = R_2(\lambda_0, \mathcal{W})R_2(\bar{\lambda}_0, \mathcal{W})$, we can obtain an expression for $R_1(\lambda_0, \mathcal{W})$ by combining two copies of (39), one with λ_0 replaced by $\bar{\lambda}_0$. Combining two matrices in the middle of the product, we eliminate one sparse matrix-vector product and one saxpy operation. Thus the cost of applying R_1 is just slightly less than twice that of applying R_2. We summarize the costs in the following table.

Table 2 Operation count for applying the operator R_1, assuming a sparse LU factorization of $Q(\lambda_0)$ is available.

8	sparse triangular solves
4	symmetric sparse Matrix vector products Mz
5	skew symmetric sparse Matrix vector products Gz
9	saxpy operations.

In general the arithmetic is complex; however, if λ_0 is real, then all operations are real.

The symplectic operators S_1 and S_2 can be applied similarly. Inverting (37) and replacing σ by $-\sigma$, we obtain

$$(\mathcal{W} + \sigma I) = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}^{-1} \begin{bmatrix} I & \sigma I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -Q(-\sigma) \\ I & 0 \end{bmatrix} \begin{bmatrix} I & \sigma M - \frac{1}{2}G \\ 0 & I \end{bmatrix}. \qquad (40)$$

Now combining (37) with (40) and replacing σ by λ_0, we get the expression

$$\begin{aligned} S_2(\lambda_0, \mathcal{W}) &= (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I) \\ &= \begin{bmatrix} I & \lambda_0 M + \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\lambda_0)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & 2\lambda_0 I \\ 0 & I \end{bmatrix} \\ &\quad \times \begin{bmatrix} 0 & -Q(-\lambda_0) \\ I & 0 \end{bmatrix} \begin{bmatrix} I & \lambda_0 M - \frac{1}{2}G \\ 0 & I \end{bmatrix}, \end{aligned} \tag{41}$$

which can be readily translated into an algorithm for applying the operator S_2. The total work for multiplying $S_2(\lambda_0, \mathcal{W})$ by a vector is summarized in the next Table.

Table 3 Operation count for applying the operator S_2, assuming a sparse LU factorization of $Q(\lambda_0)$ is available.

2	sparse triangular solves
3	symmetric sparse Matrix vector products Mz
3	skew symmetric sparse Matrix vector products Gz
1	symmetric sparse Matrix vector product Kz
7	saxpy Operations.

We have counted the operation $Q(-\lambda_0)v$ as three matrix-vector products, one each by M, G, and K, and two saxpy operations. The work can be decreased by computing $Q(-\lambda_0)$ in advance and storing it, if there is sufficient storage space. If λ_0 is real, then all operations are real.

Juxtaposing two copies of (41), we obtain the following expression for $S_1(\lambda_0, \mathcal{W})$

$$\begin{aligned} S_1(\lambda_0, \mathcal{W}) &= (\mathcal{W} - \lambda_0 I)^{-1}(\mathcal{W} + \lambda_0 I)(\mathcal{W} - \bar{\lambda}_0 I)^{-1}(\mathcal{W} + \bar{\lambda}_0 I) \\ &= \begin{bmatrix} I & \lambda_0 M + \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\lambda_0)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & 2\lambda_0 I \\ 0 & I \end{bmatrix} \\ &\quad \times \begin{bmatrix} 0 & -Q(-\lambda_0) \\ I & 0 \end{bmatrix} \begin{bmatrix} I & 2\Re(\lambda_0) M \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\bar{\lambda}_0)^{-1} & 0 \end{bmatrix} \\ &\quad \times \begin{bmatrix} I & 2\bar{\lambda}_0 I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -Q(-\bar{\lambda}_0) \\ I & 0 \end{bmatrix} \begin{bmatrix} I & \bar{\lambda}_0 M - \frac{1}{2}G \\ 0 & I \end{bmatrix}, \end{aligned} \tag{42}$$

which can be translated into an algorithm for applying the operator S_1. The total work for multiplying $S_1(\lambda_0, \mathcal{W})$ by a vector is summarized in the next Table.

Table 4 Operation count for applying the operator S_1, assuming a sparse LU factorization of $Q(\lambda_0)$ is available.

4	sparse triangular solves
5	symmetric sparse matrix-vector products Mz
4	skew symmetric sparse matrix-vector products Gz
2	symmetric sparse matrix-vector products Kz
11	saxpy operations.

In general the arithmetic is complex; however, if λ_0 is real, then all operations are real.

Typically the triangular *LU* factors are sparse, but not nearly as sparse as the original matrices. Thus the most expensive operations listed in Tables 1–4 are the sparse triangular solves. The matrix-vector products are cheaper, but still significant. The saxpy operations are relatively insignificant. Bearing these facts in mind and comparing Tables 2 and 4, we conclude that the symplectic operator S_1 can be applied significantly less expensively than R_1 can.

4.2 Optimal control problems

Consider the SHH pencil

$$\alpha \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix} - \beta \begin{bmatrix} A & -BB^T \\ C^TC & -A^T \end{bmatrix} \tag{43}$$

from the linear quadratic optimal control problems for descriptor systems, and consider the decomposition of $\mathcal{N}$ given by (5). We have

$$\begin{aligned} R_1 &= R_1(\lambda_0, \mathcal{H}, \mathcal{N}, \mathcal{Z}_1, \mathcal{Z}_2) \\ &= \mathcal{Z}_2(\mathcal{H} - \lambda_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} + \lambda_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} - \bar{\lambda}_0\mathcal{N})^{-1}\mathcal{N}(\mathcal{H} + \bar{\lambda}_0\mathcal{N})^{-1}\mathcal{Z}_1 \end{aligned}$$

from (22). In this case each of the terms $(\mathcal{H} - \sigma\mathcal{N})^{-1}$ can be factored as

$$(\mathcal{H} - \sigma\mathcal{N})^{-1} = \begin{bmatrix} A - \sigma E & -BB^T \\ C^TC & -\left(A^T + \sigma E^T\right) \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} F^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & BB^T \\ 0 & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^TC & I \end{bmatrix} \begin{bmatrix} F^{-1} & 0 \\ 0 & I \end{bmatrix}, \tag{44}$$

where $F = A - \sigma E$ and $D = -\left(A^T + \sigma E^T\right) + C^TCF^{-1}BB^T$. To apply this operator we will certainly need an *LU* factorization of $A - \sigma E$. We will also need an *LU* factorization of $A^T + \sigma E^T$, which we will use together with the Sherman/Morrison/Woodbury formula [16] to evaluate D^{-1}.

We have to apply (44) with $\sigma = \lambda_0$, $\bar{\lambda}_0$, $-\lambda_0$, and $-\bar{\lambda}_0$. If we have sparse *LU* factorizations associated with one choice of σ, then we automatically get the other factorizations. For example, if we have *LU* factorizations $A - \lambda_0 E = L_1U_1$ and $A^T + \lambda_0 E^T = L_2U_2$, then we also have

$$\begin{array}{rclrcl} A - \bar{\lambda}_0 E &=& \bar{L}_1\bar{U}_1, & A^T + \bar{\lambda}_0 E^T &=& \bar{L}_2\bar{U}_2, \\ A + \lambda_0 E &=& U_2^T L_2^T, & A^T - \lambda_0 E^T &=& U_1^T L_1^T, \\ A + \bar{\lambda}_0 E &=& \bar{U}_2^T \bar{L}_2^T, & A^T + \bar{\lambda}_0 E^T &=& \bar{U}_1^T \bar{L}_1^T. \end{array} \tag{45}$$

To apply D^{-1} efficiently, we exploit the fact that the term $C^TCF^{-1}BB^T$ has low rank. Recalling that m and p are the number of columns in B and C^T, respectively, let us assume that $p \leq m$. (Otherwise exchange the roles of B and C.) Then, with $A - \sigma E = T_1V_1$,

$-\left(A^T + \sigma E^T\right) = T_2 V_2$ being any of the factorizations listed above, we can apply the Sherman/Morrison/Woodbury formula [16] to obtain

$$\begin{aligned} D^{-1} &= (T_2V_2 + C^TCV_1^{-1}T_1^{-1}BB^T)^{-1} \\ &= V_2^{-1}(I - T_2^{-1}C^TN^{-1}CV_1^{-1}T_1^{-1}BB^TV_2^{-1})T_2^{-1}, \end{aligned} \tag{46}$$

where $N = I + CV_1^{-1}T_1^{-1}BB^TV_2^{-1}T_2^{-1}C^T$ is by assumption a very small matrix ($p \times p$) and typically full. The cost to compute N consists of $4p$ triangular solves plus $4nmp + 2mp^2$ flops (much less if B is sparse). An additional $2p^3$ flops are needed to invert this small matrix. This part of the computation must be done four times, once for each choice of σ.

With N^{-1} available we see from (46) that the cost of multiplying D^{-1} by a vector is essentially that of six sparse triangular solves. The other operations, such as multiplication of B by a vector, are relatively insignificant if B and C are sparse. Thus the cost of applying $(\mathcal{H} - \sigma\mathcal{N})^{-1}$ in (44) is 10 sparse triangular solves plus small change, and the cost of applying R_1 in (22) is 40 sparse triangular solves plus change. Similarly, the cost of applying S_1 (30) is a bit more than 20 sparse triangular solves.

4.3 Skew-Hamiltonian versus symplectic operators

We have already discussed some of the advantages and disadvantages of the two rational transformations that lead to skew-Hamiltonian and symplectic operators, respectively.

The advantage of the skew-Hamiltonian operator is that we can apply the implicitly restarted Arnoldi method, while for the symplectic operator we have to use special symplectic Lanczos methods with all their possible stability problems due to breakdowns or near breakdowns. Due to the possible breakdowns, these latter methods are also more difficult to implement in practice, see [14]. Furthermore, as we have discussed, the symplectic operator cannot be used when the target shift is close to or on the imaginary axis, which is a frequent situation in optimal control problems.

But the symplectic operators also have advantages, since the cost of applying the operator is roughly half of that for the skew Hamiltonian operator, and the method is also applicable for complex pencils. It should be noted, though, that in the symplectic Lanczos method we need to apply both the operator and its inverse at each step.

Taking this comparison into account, we have implemented the implicitly restarted Arnoldi method based on the skew-Hamiltonian operator for its simplicity and numerical robustness, and since we can also use it with shifts near or on the imaginary axis.

5 Implementation of the skew-Hamiltonian Arnoldi Method

We implemented a modified version of the implicitly restarted Arnoldi (IRA) method [37] that applies the skew-Hamiltonian operator R_1 (22). Since our method is a skew-Hamiltonian, isotropic, implicitly-restarted Arnoldi method, we call it SHIRA.

Instead of a standard Arnoldi step we use the isotropic Arnoldi step (23), (24) with $A = R_1$. In exact arithmetic the coefficients t_{ij} should all be zero, but in practice roundoff errors cause them to be nonzero. By subtracting out the tiny components $Jq_i t_{ij}$, we ensure that the spaces $\operatorname{span}\{q_1, \ldots, q_j\}$ are isotropic to working precision. For accuracy we use the modified Gram-Schmidt method [16] to compute the coefficients h_{ij} and t_{ij}, rather than applying the formulas (23) literally. Finally, since we also want the vectors to be orthonormal to working precision, we apply reorthogonalization [12] in the computation of the h_{ij}. That is, we do two orthogonalization sweeps on $q_1, \ldots, q_j$. After j steps of the isotropic Arnoldi process we have vectors $q_1, \ldots, q_{j+1}$ satisfying

$$AQ_j = Q_j H_j + JQ_j T_j + q_{j+1} h_{j+1,j} e_j^T, \tag{47}$$

where $Q_j = [q_1 \cdots q_j]$, and H_j and T_j are matrices of coefficients. Our method ignores T_j and so depends on the fact that the elements of T_j are tiny. Our implementation of IRA is standard [37]; the shifts are chosen to select for the eigenvalues of A of largest magnitude. Each implicit restart, i.e. each iteration of the IRA method, produces a new configuration of the form (47) with a different starting vector q_1 and, typically, a smaller $h_{j+1,j}$. After several restarts $h_{j+1,j}$ becomes negligible, and we have, up to roundoff errors,

$$AQ_j = Q_j H_j.$$

The eigenvalues of H_j are j of the eigenvalues of $A = R_1(\lambda_0, \mathcal{W})$, and they are typically the j largest in magnitude. If we actually want fewer than j eigenvalues, we can monitor $h_{k+1,k}$ for $k \leq j$. If we stop when $h_{k+1,k}$ is negligible, we get k eigenvalues.

5.1 Eigenvalue computation

The Arnoldi process yields eigenvalues of $R_1(\lambda_0, \mathcal{W})$, but we actually want eigenvalues of the Hamiltonian matrix $\mathcal{W}$. Each eigenvalue μ of R_1 corresponds to two eigenvalues $\pm\lambda$ of $\mathcal{W}$ satisfying $(\lambda^2 - \lambda_0^2)(\lambda^2 - \bar{\lambda}_0^2) = 1/\mu$. A seemingly straightforward approach is to solve the quadratic equation

$$(\nu - \lambda_0^2)(\nu - \bar{\lambda}_0^2) = 1/\mu \tag{48}$$

for ν, then compute $\pm\sqrt{\nu}$ to get the eigenvalues. Unfortunately (48) has two solutions, only one of which corresponds to eigenvalues of $\mathcal{W}$. Thus one is faced with deciding which ν is the correct one.

We have adopted a different approach, which dodges this decision and also allows us to make a final test of the backward stability of the result. Once $h_{k+1,k}$ is negligible, the space $\operatorname{span}\{q_1, \ldots, q_k\}$ is, up to roundoff errors, an invariant subspace under R_1. Normally it is also invariant under $\mathcal{W}^2$ (but not under $\mathcal{W}$). We calculate the Ritz values of $\mathcal{W}^2$ with respect to the space $\operatorname{span}\{q_1, \ldots, q_k\}$; that is, we calculate the eigenvalues μ_i of $B = Q_k^T \mathcal{W}^2 Q_k$. Then $\pm\sqrt{\mu_i}$ are the eigenvalues of $\mathcal{W}$ that we seek.

As a byproduct of the computation of B, we obtain the vectors $\mathcal{W}^2 Q_k$, which we use to compute the residual

$$\left\| \mathcal{W}^2 Q_k - Q_k B \right\|_F \tag{49}$$

and thereby check whether or not span$\{q_1, \ldots, q_k\}$ really is invariant under $\mathcal{W}^2$. This test is necessary because it can happen that a space that is invariant under R_1 fails to be invariant under $\mathcal{W}^2$. If λ_0 is chosen so unfortunately that two distinct eigenvalues of $\mathcal{W}^2$ are mapped to the same eigenvalue of R_1, then R_1 will have a four-dimensional eigenspace spanned by two two-dimensional eigenspaces of $\mathcal{W}^2$. If the Arnoldi process picks up only one vector from this space, it will typically not be an eigenvector of $\mathcal{W}^2$, and the space span$\{q_1, \ldots, q_k\}$ will not be invariant under $\mathcal{W}^2$.

It is interesting to note that the practical need for such a stability test is related to our insistence upon enforcing isotropy. In principle a Krylov subspace can contain at most a one-dimensional subspace of a multi-dimensional eigenspace, so the Arnoldi process will find at most one copy of a geometrically-multiple eigenvalue. However, as is well known [38], roundoff errors will turn multiple eigenvalues into simple ones and hence we will detect different approximations of the multiple eigenvalues. This is mostly a nuisance for us, because the eigenvalues of $\mathcal{W}^2$ are all geometrically multiple, and we would rather not pay to calculate two copies of each eigenvalue. Therefore we use the isotropic Arnoldi method which enforces isotropy. It has the effect that each eigenvalue is picked up only once in practice. It has the unfortunate side effect that when a four-dimensional invariant subspace of R_1 consists of two two-dimensional invariant subspaces of $\mathcal{W}^2$, only one vector from that space is found, from which it is impossible to deduce eigenvectors of $\mathcal{W}^2$. Fortunately the merging of eigenspaces is a rare event.

Finally we should note that this approach to eigenvalue calculation requires application of the operator $\mathcal{W}^2$. In our quadratic eigenvalue applications, linearized as in (8) or (11), we have

$$\mathcal{W} = \begin{bmatrix} I & -\frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -K \\ M^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & -\frac{1}{2}G \\ 0 & I \end{bmatrix}.$$

To apply M^{-1}, we need to compute the Cholesky decomposition of M. Typically the Cholesky factor is quite sparse, and this step does not add substantially to the overall computing time. At this point in the computation we no longer need the LU factors of $Q(\lambda_0)$ (used for applying R_1), so we can use that storage space for the Cholesky factor of M.

5.2 Eigenvector Computation

In the course of the eigenvalue computation we can easily obtain the corresponding Ritz vectors, which are eigenvectors of $\mathcal{W}^2$. Each of these is a particular member of a two-dimensional eigenspace of $\mathcal{W}^2$ but in general not an eigenvector of $\mathcal{W}$. Thus, if we want eigenvectors to go with our eigenvalues, we need to do more work.

Given eigenvalues, the quickest way to obtain corresponding eigenvectors is to perform inverse iteration. In the case of the quadratic eigenvalue problem we refer back to the original form of the problem:

$$Q(\lambda)v = \lambda^2 Mv + \lambda Gv + Kv = 0.$$

For each eigenvalue λ we form a sparse LU decomposition of $Q(\lambda)$ and use the decomposition to perform one step of inverse iteration

$$v = Q(\lambda)^{-1}w.$$

As a starting vector w we use the bottom half of the $2n$-dimensional Ritz vector of $\mathcal{W}^2$ associated with λ^2. The residual $\|Q(\lambda)v\|_2 \,/\, \|v\|_2$ gives another measure of backward stability of the computation.

To compute the eigenvector associated with $-\lambda$ we exploit the relationship $Q(-\lambda) = Q(\lambda)^T$. Thus, as in (38), the LU decomposition of $Q(\lambda)$ can also be used for the computation $v = Q(-\lambda)^{-1}w$.

If the complex eigenvalue λ has eigenvector v, then $\bar{\lambda}$ has eigenvector $\bar{v}$.

In summary, the cost of computing the eigenvectors associated with a pair $\{\lambda, -\lambda\}$ or a quadruple $\{\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda}\}$ is one sparse LU decomposition plus four sparse triangular solves. If several sets of eigenvectors are wanted, they can be computed in parallel, given available processors and memory, or they can be computed sequentially, reusing the memory space for the LU decompositions.

6 Numerical Results

We applied the skew-Hamiltonian Arnoldi method SHIRA to solve numerous quadratic eigenvalue problems

$$\lambda^2 Mx + \lambda Gx + Kx = 0. \tag{50}$$

All computations were done in Matlab Version 5.2 on a Linux machine.

In one class of problems that we considered, we built matrices of order $n = m^2$ by a tensor product construction. Let B denote the $m \times m$ nilpotent Jordan block

$$B = \begin{bmatrix} 0 & & & 0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix},$$

and define $\tilde{M} = \frac{1}{6}(4I_m + B + B^T)$, $\tilde{G} = B - B^T$, and $\tilde{K} = -(2I_m - B - B^T)$. Then we set

$$\begin{aligned} M &= c_{11} I_m \otimes \tilde{M} + c_{12}\tilde{M} \otimes I_m, \\ G &= c_{21} I_m \otimes \tilde{G} + c_{22}\tilde{G} \otimes I_m, \\ K &= c_{31} I_m \otimes \tilde{K} + c_{32}\tilde{K} \otimes I_m, \end{aligned} \tag{51}$$

where the coefficients c_{ij} are positive constants. We have $M = M^T > 0$, $G = -G^T$, and $K = K^T < 0$.

Example 7. If we take $m = 10$ and

$$\begin{aligned} c_{11} &= 1.00, & c_{12} &= 1.30, \\ c_{21} &= 1.35, & c_{22} &= 1.10, \\ c_{31} &= 1.00, & c_{32} &= 1.20, \end{aligned} \tag{52}$$

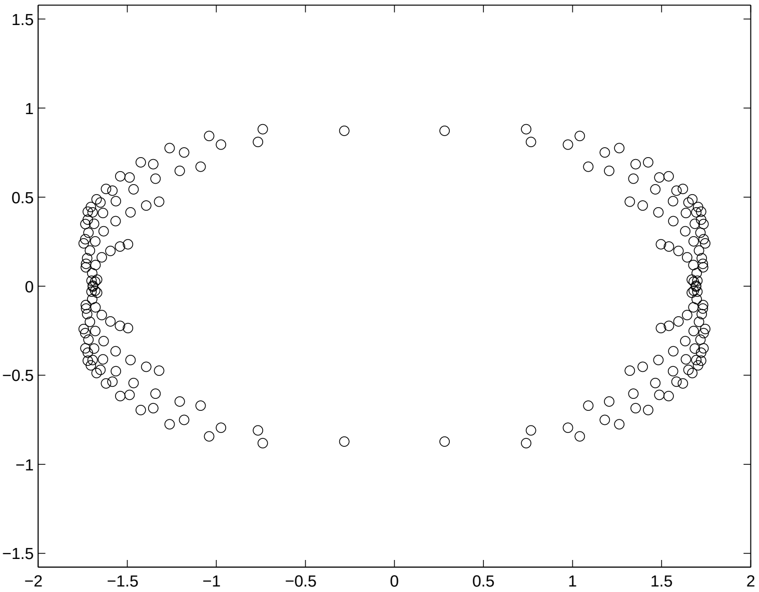

Figure 1: Eigenvalues of 100 × 100 quadratic pencil

then we obtain a 100 × 100 quadratic pencil, whose 200 eigenvalues are shown in Figure 1. These were computed by applying Matlab's `eig` command to the 200 × 200 matrix pencil (11), ignoring all structure, at a cost of 885×10^6 flops.

Suppose we want to use SHIRA to compute the 12 eigenvalues that are closest to the imaginary axis. Then, supposing that we know nothing about where the eigenvalues lie, our safest course of action is to choose a target shift λ_0 that lies on the imaginary axis. Table 5 shows the flop counts for computing these eigenvalues using three different choices of purely imaginary target. Results are given for our structured method SHIRA and a competing unstructured method, which applies IRA (in complex arithmetic) to the shifted inverted Hamiltonian operator $(\mathcal{W} - \lambda_0 I)^{-1}$, see (37). Because of the shift, this operator has no structure.

Table 5 Flop count for computing the 12 smallest eigenvalues (and associated eigenvectors) of the pencil in Example 7 using structured and unstructured methods.

	flops (10^6)	
λ_0	SHIRA	unstructured
$0.1i$	17.3	71.8
$1.0i$	11.4	26.3
$5.0i$	44.9	98.2

We see that both methods benefit from a good choice of shift, but regardless of shift, SHIRA outperforms the unstructured method by a factor of two or more. SHIRA applies IRA (in real arithmetic) to the skew-Hamiltonian operator R_1 (20) and finds the six largest eigenvalues (three complex-conjugate pairs), which correspond to three quadruplets of eigenvalues of (50). We used 10 Arnoldi steps per implicit restart. Using the shift $\lambda_0 = i$, for example, the competing method finds the six eigenvalues (in the upper half plane) that are closest to the shift, not realizing that they constitute three pairs $(\lambda, -\bar{\lambda})$. We then deduce six other eigenvalues by taking complex conjugates. Again we used 10 Arnoldi steps per implicit restart. For both methods we used a deflation tolerance of 10^{-10}. In all cases the errors in the computed eigenvalues and eigenvectors were less than 10^{-9}. □

Example 8. Consider another pencil built from matrices of the form (51), this time with $m = 5$. Use the same coefficients as in (52), except that $c_{21} = 0.1$. This results in a pencil whose smallest eigenvalues are real. The three smallest positive eigenvalues are

$$\begin{aligned} \lambda_1 &= 0.6726432397672 \\ \lambda_2 &= 0.9866442639296 \\ \lambda_3 &= 1.0689101679903. \end{aligned}$$

Using SHIRA with an appropriate tolerance and any reasonable shift (e.g. $\lambda_0 = 0$, i, or 0.5), we can compute these eigenvalues quickly, with any desired accuracy up to machine precision. The purpose of this example is to show that an unfortunate choice of shift can cause our eigenvalue computation scheme to fail. If we take $\lambda_0 = \hat{\lambda}_0 = \sqrt{(\lambda_1^2 + \lambda_2^2)/2}$, then the two eigenvalues λ_1^2 and λ_2^2 of $\mathcal{W}^2$ are mapped to the same eigenvalue of $R_1(\lambda_0, \mathcal{W})$, which then has a four-dimensional eigenspace associated with that eigenvalue. With this choice of λ_0, SHIRA is unable to calculate correct values for λ_1 and λ_2. Checking the residual (49), we find that $\|\mathcal{W}^2 Q_k - Q_k B\|_F \approx .37\, \|\mathcal{W}^2 Q_k\|_F$, indicating that the space $\text{span}\{q_1, \ldots, q_k\}$ is not invariant under $\mathcal{W}^2$. This is a red flag that tells us that our results cannot be trusted.

A simple remedy is to change λ_0. Even a tiny change suffices. For example, if we run SHIRA with $\lambda_0 = \hat{\lambda}_0 + 10^{-5}$ (and deflation tolerance 10^{-10}), we get the three smallest eigenvalues correct to twelve decimal places. The residual (49) is $\|\mathcal{W}^2 Q_k - Q_k B\|_F \approx (4 \times 10^{-11})\, \|\mathcal{W}^2 Q_k\|_F$, indicating that the subspace is invariant under $\mathcal{W}^2$ to within the desired tolerance. □

Example 9. Finally we consider a quadratic eigenvalue problem obtained by a finite element discretization of equations of elastic deformation of an anisotropic material [18, 20, 32]. The matrices have dimension 2223. Suppose we wish to find the twelve eigenvalues closest to the imaginary axis. It is known a priori that the eigenvalues lie near the real axis, so it makes sense to use a real target shift. In fact, the six smallest eigenvalues in the right half-plane are

$$\begin{aligned}
\lambda_1 &= 0.96269644895 \\
\lambda_2 &= 0.98250961158 + 0.00066849814i \\
\lambda_3 &= 0.98250961158 - 0.00066849814i \\
\lambda_4 &= 1.35421843051 \\
\lambda_5 &= 1.39562564903 \\
\lambda_6 &= 1.49830518846.
\end{aligned}$$

The flops needed to compute these eigenvalues by SHIRA and the unstructured method using various choices of target shift are given in Table 6.

Table 6 Flop count for computing smallest 12 eigenvalues of quadratic pencil of Example 9 using structured and unstructured methods.

	flops (10^7)	
λ_0	SHIRA	unstructured
0	32.6	140.1
0.3	32.6	79.6
0.6	28.4	69.8
0.9	28.4	50.7
1.2	19.8	31.5

Again we used a deflation tolerance of 10^{-10}. We see that SHIRA can benefit from a good choice of λ_0 but also does well if a good shift is not known. In particular it performs well even for the poor but safe choice $\lambda_0 = 0$. In each case SHIRA computes the six largest eigenvalues of $R_1(\lambda_0, \mathcal{W})$, which turn out to be one complex pair and four real eigenvalues. These yield twelve eigenvalues of $\mathcal{W}$, one complex quadruplet and four real $\{\lambda, -\lambda\}$ pairs. In all of these runs we used nine Arnoldi steps per implicit restart.

In contrast with SHIRA, the unstuctured method is highly dependent on a good choice of shift. Given a good shift, such as $\lambda_0 = 1.2$, the unstructured method was nearly competitive with SHIRA. It computed the six positive eigenvalues closest to λ_0, which turned out to be the eigenvalues that we wanted. We then deduced the six negative eigenvalues $-\lambda_1, \ldots, -\lambda_6$ from the structure. We used nine Arnoldi steps per implicit restart.

The problem with the unstructured approach is that if we choose a target that is too large, we might miss the smallest eigenvalues. On the other hand, if we take a shift that is too close to the origin, we end up computing some of the left half-plane eigenvalues explicitly. For example, in the case $\lambda_0 = 0.3$, λ_6 is not among the six smallest eigenvalues of the shifted operator. We had to compute nine eigenvalues in order to find $\lambda_1, \ldots, \lambda_6$; we also got $-\lambda_1$, $-\lambda_2$, and $-\lambda_3$. We took twelve Arnoldi steps per restart.

Using the shift $\lambda_0 = 0$, the safest choice, we have to choose all twelve eigenvalues explicitly. We used fifteen Arnoldi steps per restart. As Table 6 shows, many more flops were needed in this case than in all other cases. □

7 Conclusion

We have discussed structure preserving shift-and-invert Krylov subspace methods for the computation of a few eigenvalues and eigenvectors of large, sparse skew-Hamiltonian/Hamiltonian pencils. We have demonstrated that a skew-Hamiltonian shift-and-invert implicitly-restarted Arnoldi method can speed up the computation of desired eigenvalues in the interior of the spectrum significantly. Furthermore, by this approach it is guaranteed that the computed spectrum has the correct symmetry structure.

References

[1] G. Ammar, C. Mehl, and V. Mehrmann. Schur-like forms for matrix Lie groups, Lie algebras, and Jordan algebras. *Linear Algebra Appl.*, 287:11–39, 1999.

[2] G.S. Ammar and V. Mehrmann. On Hamiltonian and symplectic Hessenberg forms. *Linear Algebra Appl.*, 149:55–72, 1991.

[3] P. Benner, R. Byers, V. Mehrmann, and H. Xu. Numerical computation of deflating subspaces of embedded Hamiltonian pencils. Technical Report 99-15, Sondeforschungsbereich 393, Numerische Simulation auf massiv parallelen Rechnern, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, Germany, 1999. Available online from `http://www.tu-chemnitz.de/sfb393/sfb99pr.html`.

[4] P. Benner, R. Byers, V. Mehrmann, and H. Xu. Numerical methods for linear quadratic and H_∞ control problems. In G. Picci und D.S. Gillian, editor, *Dynamical Systems, Control, Coding, Computer Vision. Progress in Systems and Control Theory, Vol. 25*, pages 203–222, Basel, 1999. Birkhäuser Verlag.

[5] P. Benner and H. Faßbender. An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem. *Linear Algebra Appl.*, 263:75–111, 1997.

[6] P. Benner and H. Faßbender. The symplectic eigenvalue problem, the butterfly form, the *SR* algorithm, and the Lanczos method. *Linear Algebra Appl.*, 275/276:19–47, 1998.

[7] P. Benner, V. Mehrmann, and H. Xu. Perturbation analysis of the eigenvalue problem of a formal matrix product. In Preparation.

[8] P. Benner, V. Mehrmann, and H. Xu. A new method for computing the stable invariant subspace of a real Hamiltonian matrix. *J. Comput. Appl. Math.*, 86:17–43, 1997.

[9] P. Benner, V. Mehrmann, and H. Xu. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numer. Math.*, 78(3):329–358, 1998.

[10] P. Benner, V. Mehrmann, and H. Xu. A note on the numerical solution of complex Hamiltonian and skew-Hamiltonian eigenvalue problems. *Electr. Trans. Num. Anal.*, 8:115–126, 1999.

[11] R. Byers, C. He, and V. Mehrmann. Where is the nearest non-regular pencil. *Linear Algebra Appl.*, 285:81–105, 1998.

[12] J. Daniel, W.B. Gragg, L. Kaufman, and G.W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math. Comp.*, 30:772–795, 1976.

[13] E.G. D'yakunov and A.V. Knyazev. The group iterative method for finding low-order eigenvalues. *Vestnik Mosk. Univ., Vychisl. Mat. Kibern.*, Ser. XV, No. 2:29–34, 1982.

[14] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Comput.*, 14:137–158, 1993.

[15] R.W. Freund. Transpose-free quasi-minimal residual methods for non-hermitian linear systems. In G. Golub, A. Greenbaum, and M. Luskin, editors, *Recent Advances in Iterative Methods. The IMA Volumes in Mathematics and its Applications, Vol. 60*, pages 69–94. Springer Verlag, New York, 1994.

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

[17] A. S. Hodel and K.R. Poolla. Heuristic approaches to the solution of very large sparse Lyapunov and algebraic Riccati equations. In *Proc. 27th IEEE Conf. Decis. Cont., Austin, TX*, pages 2217–2222, 1988.

[18] V.A. Kozlov, V.G. Maz'ya, and J. Roßmann. Spectral properties of operator pencils generated by elliptic boundary value problems for the Lamé system. *Rostocker Math. Kolloq.*, 51:5–24, 1997.

[19] P. Lancaster. Strongly stable gyroscopic systems. *Electr. J. Linear Algebra*, 5:53–66, 1999.

[20] D. Leguillon. Computation of 3d-singularities in elasticity. In M. Costabel et al., editor, *Boundary value problems and integral equations in nonsmooth domains*, volume 167 of *Lect. Notes Pure Appl. Math.*, pages 161–170, New York, 1995. Marcel Dekker. Proceedings of the conference, held at the CIRM, Luminy, France, May 3-7, 1993.

[21] W.-W. Lin and V. Mehrmann und H. Xu. Canonical forms for Hamiltonian and symplectic matrices and pencils. *Linear Algebra Appl.*, 301-303:469–533, 1999.

[22] K. Meerbergen. Locking and restarting quadratic eigenvalue solvers. Report RAL-TR-1999-011, CLRC, Rutherford Appleton Laboratory, Dept. of Comp. and Inf., Atlas Centre, Oxon OX11 0QX, GB, 1999.

[23] C. Mehl. *Compatible Lie and Jordan algebras and applications to structured matrices and pencils.* PhD thesis, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz (FRG), 1998.

[24] C. Mehl. Condensed forms for skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Matrix Anal. Appl.*, 21:454–476, 1999.

[25] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution.* Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[26] V. Mehrmann. A step toward a unified treatment of continuous and discrete time control problems. *Linear Algebra Appl.*, 241–243:749–779, 1996.

[27] A. Meyer. Die simultane Iteration zur Eigenwertberechnung für beliebige endlichdimensionale Operatoren und die Anwendung auf Schwingungsprobleme. Report Wissenschaftliche Informationen 36, Technische Hochschule Karl-Marx-Stadt (now Technische Universität Chemnitz), Fakultät für Mathematik, 1982.

[28] T. Penzl. A cyclic low rank Smith method for large, sparse Lyapunov equations with applications in model reduction and optimal control. Technical Report SFB393/98-6, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, FRG, 1998. Available from `http://www.tu-chemnitz.de/sfb393/sfb98pr.html`.

[29] I.G. Rosen and C. Wang. A multi–level technique for the approximate solution of operator Lyapunov and algebraic Riccati equations. *SIAM J. Numer. Anal.*, 32(2):514–541, 1995.

[30] Y. Saad. *Numerical methods for large eigenvalue problems.* Manchester University Press, Manchester, M13 9PL, UK, 1992.

[31] R. Sandhu and K. Pister. A variational principle for linear coupled field problems in continuum mechanics. *Internat. J. Eng. Sci.*, 8:989–999, 1970.

[32] H. Schmitz, K. Volk, and W. L. Wendland. On three-dimensional singularities of elastic fields near vertices. *Numer. Methods Partial Differ. Equations*, 9:323–337, 1993.

[33] H.R. Schwarz. *Methode der finiten Elemente.* Teubner, Stuttgart, 1984.

[34] B. Simon, J. Wu, O. Zienkiewicz, and D. Paul. Evaluation of u-w and u-p finite element methods for the dynamic response of sturated porous media using one-dimensional models. *Internat. J. Numer. Anal. Methods Geomech.*, 10:461–482, 1986.

[35] S.I. Solov'ev. The finite-element method for symmetric eigenvalue nonlinear problems. *Comp. Math. and Math. Physics*, 37(11):1269–1276, 1997. Translated from Zhurnal Vyschislitel'noi Matematiki i Matematicheskoi Fiziki.

[36] S.I. Solov'ev. Convergence of the modified subspace iteration method for nonlinear eigenvalue problems. Preprint SFB393/99-35, Sonderforschungsbereich 393, Numerische Simulation auf massiv parallelen Rechnern, Fakultät für Mathematik, TU Chemnitz, D-09107 Chemnitz, Germany, 1999.

[37] D.C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13:357–385, 1992.

[38] G.W. Stewart and J.-G. Sun. *Matrix Perturbation Theory.* Academic Press, New York, 1990.

[39] F. Tisseur. Backward error analysis of polynomial eigenvalue problems. Numerical Analysis Report 332, Manchester Centre for Computational Mathematics, Department of Mathematics, University of Manchester, Manchester M13 9Pl, GB, 1998.

[40] C.F. Van Loan. A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix. *Linear Algebra Appl.*, 61:233–251, 1984.