

## SHIRRA: A REFINED VARIANT OF SHIRA FOR THE SKEW-HAMILTONIAN/HAMILTONIAN (SHH) PENCIL EIGENVALUE PROBLEM

Zhongxiao Jia and Yuquan Sun

**Abstract.** Combining the Skew-Hamiltonian Isotropic implicitly Restarted Arnoldi algorithm (SHIRA) due to Mehrmann and Waktins and the refined projection principle proposed by the first author, we present a Skew-Hamiltonian Isotropic implicitly Restarted Refined Arnoldi algorithm (SHIRRA) for the skew-Hamiltonian/Hamiltonian (SHH) pencil eigenvalue problem. Within SHIRRA, we propose new shifts, called refined shifts, that are theoretically better and numerically more efficient than the exact shifts used within SHIRA. Numerical examples illustrate the efficiency and superiority of SHIRRA.

### 1. INTRODUCTION

Consider the computation of several eigenvalues of the large quadratic eigenvalue problem (QEP)

$$(1) \quad \lambda^2 Mx + \lambda Gx + Kx = 0, x \neq 0,$$

where  $M = M^T$  is positive definite, and  $G = -G^T$ ,  $K = K^T$  are all  $n \times n$  real matrices.

The QEPs arise in many applications, in which  $M$  is often a mass matrix and  $K$  is a stiffness matrix; see, e.g., [2]. We are interested in a few eigenvalues nearest to a target  $\sigma$ . It is well known [11] that the spectrum of (1) has special structure and the eigenvalues occur in either quadruplets  $(\lambda, \bar{\lambda}, -\lambda, -\bar{\lambda})$  or real and purely imaginary pairs  $(\lambda, -\lambda)$ .

---

Received April 22, 2012, accepted July 11, 2012.

Communicated by Wen-Wei Lin.

2010 *Mathematics Subject Classification*: 65F15.

*Key words and phrases*: Refined projection, SHH pencil, Quadratic eigenvalue problem, Ritz value, Refined eigenvector approximation, Implicit restart, Refined shifts, Exact shifts.

Supported by National Basic Research Program of China 2011CB302400 and the National Science Foundation of China (No. 11071140).

A commonly used approach is to transform (1) into a generalized eigenvalue problem, e.g.,

$$(2) \quad \lambda \begin{bmatrix} M & G \\ 0 & M \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} 0 & -K \\ M & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0$$

with  $y = \lambda x$  and then solve it by a suitable numerical method. For (2), its mathematical theory has been well established and various numerical methods have been developed [12, 14].

Mehrmman and Watkins [11] have proposed the Skew-Hamiltonian Isotropic implicitly Restarted Arnoldi algorithm (SHIRA) for the Skew-Hamiltonian/Hamiltonian (SHH) matrix pencil eigenvalue problem that is reduced to a Hamiltonian eigenvalue problem. It is an orthogonal projection method and uses the isotropic Arnoldi process to generate an orthonormal and symplectic basis of an isotropic Krylov subspace. The algorithm is structure preserving and ensures that the computed spectrum has the correct eigenvalue symmetry. The algorithm exploits the implicit restarting technique [13]. Recently, Benner, Fassbender and Stoll [1] have proposed an explicit Krylov–Schur like restarting technique that is applied within the symplectic Lanczos algorithm for the Hamiltonian eigenvalue problem. There, restarting is based on *SR* algorithm and a purging and locking strategy is implemented to improve the convergence of the symplectic Lanczos algorithm.

A general convergence theory for projection methods including the Arnoldi method can be found in [5, 9, 10] and also [14, 15]. For a sequence of projection subspaces containing increasingly accurate approximations to the eigenvectors associated with the desired eigenvalues, it has been shown that there exist Ritz values that converge to the desired eigenvalues, while the corresponding Ritz vectors may exhibit irregular convergence behavior and even may fail to converge. To correct this deficiency, the first author has proposed and developed a class of refined projection methods, also called the refined Rayleigh–Ritz method, in several papers, e.g., [6, 8]. For a systematic account, see [14, 15].

Applying the implicit restarting technique [13], the first author developed Implicitly Restarted Refined Arnoldi algorithm (IRRA) [7], where new shifts, called refined shifts, are proposed for use within each refined algorithm. The refined shifts are theoretically better than the corresponding exact shifts used in implicitly restarted Arnoldi algorithm. Extensive numerical experiments have illustrated that IRRA is often considerably more efficient than the Implicitly Restarted Arnoldi (IRA) algorithm, whose Matlab function is `eigs`.

In this paper, based on the refined projection principle [6], we present a refined variant of SHIRA. The refined method extracts certain refined approximate eigenvectors, called refined Ritz vectors, in the sense of residual minimizations of approximate eigenpairs. Applying the implicit restarting technique, we develop a Skew-Hamiltonian

Isotropic implicitly restarted Refined Arnoldi algorithm (SHIRRA). Within SHIRRA, we propose new shifts, called refined shifts, that are shown to be theoretically better and numerically more efficient than the exact shifts used within SHIRA. In order to design comparable stopping criteria for SHIRRA and SHIRA in numerical experiments, we derive some residual relationship between Ritz vectors and the invariant subspace spanned by them. We develop the Matlab code of SHIRRA and make numerical experiments to show the efficiency and superiority of SHIRRA to SHIRA.

The paper is organized as follows. In Section 2 we briefly review SHIRA. In Section 3 we propose a skew-Hamiltonian isotropic refined Arnoldi method and develop SHIRRA with the refined shifts suggested. We discuss some implementational details in Section 4. In Section 5, we report numerical experiments to show the efficiency and superiority of SHIRRA to SHIRA. Finally, we conclude the paper in Section 6.

Throughout the paper, we denote by  $\|\cdot\|$  the Euclidean vector norm and the induced spectral norm of a matrix, by  $\|\cdot\|_F$  the Frobenius norm and by the superscript  $T$  the transpose of a matrix or vector, respectively. The skew-symmetric matrix  $J$  is defined as

$$(3) \quad J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}.$$

## 2. REVIEW OF SHIRA

We briefly review SHIRA in [11]. Let

$$N = \begin{bmatrix} M & G \\ 0 & M \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -K \\ M & 0 \end{bmatrix}.$$

Then  $N$  is skew-Hamiltonian,  $U$  is Hamiltonian and (2) becomes

$$(4) \quad \lambda Nz - Uz = 0,$$

which is a SHH matrix pencil eigenvalue problem.

Let  $N$  be given in the factored form  $N = Z_1 Z_2$  with

$$(5) \quad Z_1 = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}, \quad Z_2 = \begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix}.$$

Then  $Z_2^T J = J Z_1$  and  $\lambda N - U$  is equivalent to  $\lambda I - W$ , where  $W = Z_1^{-1} U Z_2^{-1}$  is Hamiltonian. Suppose that some eigenvalues of  $W$  nearest to a target point  $\sigma$  are of interest. Then Mehrmann and Watkins [11] suggest a rational transformation with four targets  $(\sigma, -\sigma, \bar{\sigma}, -\bar{\sigma})$  given by

$$R_1(\sigma, W) = (W - \sigma I)^{-1} (W - \bar{\sigma} I)^{-1} (W + \sigma I)^{-1} (W + \bar{\sigma} I)^{-1}.$$

If the target  $\sigma$  is either real or purely imaginary, they propose to use a simpler transformation

$$R_2(\sigma, W) = (W - \sigma I)^{-1}(W + \sigma I)^{-1}.$$

If the matrix  $W$  is real and Hamiltonian, then both  $R_1(\sigma, W)$  and  $R_2(\sigma, W)$  are real and skew-Hamiltonian [11]. Hence we can get the eigenvalues near these targets if standard iterative methods [12, 13] are applied to  $R_1(\sigma, W)$  or  $R_2(\sigma, W)$ .

Assume  $A \in \mathcal{R}^{2n \times 2n}$  to be skew-Hamiltonian. Then it is shown [16] that there exists a column orthonormal  $Q \in \mathcal{R}^{2n \times n}$  such that  $[Q, JQ]$  is orthogonal and symplectic and

$$(6) \quad [Q, JQ]^T A [Q, JQ] = \begin{bmatrix} H & F \\ 0 & H^T \end{bmatrix}$$

with  $F^T = -F$  and  $H$  upper Hessenberg. For each double eigenvalue of  $A$ , one copy resides in  $H$  and the other copy is in  $H^T$ . So the spectral information of  $A$  will not be lost when we only deal with  $H$ . Define the Krylov subspace

$$\mathcal{K}_m(A, q_1) = \text{span}\{q_1, Aq_1, \dots, A^{m-1}q_1\},$$

starting with a unit length vector  $q_1$ . This is an isotropic subspace and an orthonormal and symplectic basis  $\{q_j\}_{j=1}^m$  of it can be generated by the  $m$ -step isotropic Arnoldi process

$$(7) \quad AQ_m = Q_m H_m + JQ_m T_m + q_{m+1} h_{m+1m} e_m^T = Q_{m+1} \tilde{H}_m,$$

where  $Q_m = (q_1, \dots, q_m)$ ,  $T_m = 0$  is zero in exact arithmetic,  $H_m$  is an  $m \times m$  upper Hessenberg matrix,  $e_m$  is the  $m$ -th coordinate vector of dimension  $m$ , and  $\tilde{H}_m$  is the  $(m+1) \times m$  augmented matrix of  $H_m$  with the last row being  $(0, \dots, 0, h_{m+1m})$ .

From (7) and the definition of  $J$ , we get

$$(8) \quad Q_m^T A Q_m = H_m,$$

$$(9) \quad [Q_m, JQ_m]^T A [Q_m, JQ_m] = \begin{bmatrix} H_m & F_m \\ & H_m^T \end{bmatrix}.$$

SHIRA computes the eigenvalues  $\tilde{\mu}_i$ ,  $i = 1, 2, \dots, m$ , of  $H_m$  and select  $k$  ones among them to approximate the desired  $k$  eigenvalues of  $A$ . This way computes some eigenvalues of  $A$  in duplicate and the other copy resides in  $H_m^T$ . If the  $k$  Ritz values converge, one will get simultaneously  $2k$  eigenvalues of  $A$ .

### 3. SHIRRA AND A REFINED SHIFTS SCHEME

As shown in [7, 13], the selection of shifts is one of the keys for the success of an implicitly restarted algorithm. For SHIRA, Mehrmann and Watkins take those unwanted  $m-k$  eigenvalues (Ritz values) of  $H_m$  as shifts, called exact shifts. Combining SHIRA

and the refined Arnoldi method [7], in this section we present a refined variant of SHIRA and develop a Skew-Hamiltonian Isotropic implicitly Restarted Refined Arnoldi algorithm (SHIRRA) with certain refined shifts proposed. We will show that the refined shifts are theoretically better and numerically more efficient than the exact shifts.

Suppose  $y_i$ 's are the eigenvectors associated with the desired  $\mu_i, i = 1, 2, \dots, k$  of  $A = R_1(\sigma, W)$  or  $R_2(\sigma, W)$ . If an  $m$ -step isotropic Arnoldi process is run, then the refined Arnoldi method works as follows: It first computes the eigenvalues (Ritz values) of  $H_m$  and among them selects  $k$  Ritz values  $\tilde{\mu}_i, i = 1, 2, \dots, k$  as approximations to the desired  $\mu_i$ . Then for each  $\tilde{\mu}_i, i = 1, 2, \dots, k$  the structure preserving refined Arnoldi method seeks a unit length vector  $u_i \in \mathcal{K}_m(A, q_1)$  satisfying the optimality

$$(10) \quad \|(A - \tilde{\mu}_i I)u_i\| = \min_{\|u\|=1, u \in \mathcal{K}_m(A, q_1)} \|(A - \tilde{\mu}_i I)u\|$$

and uses it to approximate  $y_i$ .  $u_i$  is called a refined eigenvector approximation.

Let  $z_i$  be the right singular vector of the  $(m+1) \times m$  matrix  $\tilde{H}_m - \tilde{\mu}_i \tilde{I}_m$  associated with its smallest singular value  $\sigma_{\min, i}$ , where  $\tilde{I}_m$  is the same as the  $m \times m$  identity matrix augmented with an additional zero row. Then it is shown [6] that

$$(11) \quad u_i = Q_m z_i,$$

$$(12) \quad \|(A - \tilde{\mu}_i I)u_i\| = \sigma_{\min, i}.$$

Let  $\varphi_i, i = 1, 2, \dots, k$  be the Ritz vectors of  $A$  associated with the  $\tilde{\mu}_i$ 's. Next we develop SHIRRA. As in SHIRA, the implicit restarting technique is directly applicable. So we are concerned with a reasonable selection of shifts involved. For IRRA, based on the refined eigenvector approximations, the first author [7] has proposed the refined shifts that are theoretically better and numerically more efficient than the exact shifts [13] for IRA. In the same spirit, we can adapt the refined shifts scheme to SHIRRA.

As done in [7], we make the following orthogonal direct sum decompositions:

$$\begin{aligned} \mathcal{K}_m(A, q_1) &= \text{span}\{u_1, \dots, u_k\} \oplus \text{span}\{u_1, \dots, u_k\}^\perp \\ &= \text{span}\{\varphi_1, \dots, \varphi_k\} \oplus \text{span}\{\varphi_1, \dots, \varphi_k\}^\perp, \end{aligned}$$

Note that SHIRA uses the unwanted Ritz values  $\tilde{\mu}_j, j = k+1, \dots, m$  as shifts, called the exact shifts. As is done in [7], it can be verified that the exact shifts are the Ritz values of  $A$  with respect to the deflated subspace  $\text{span}\{\varphi_1, \dots, \varphi_k\}^\perp$ , which is the orthogonal complement of  $\text{span}\{\varphi_1, \dots, \varphi_k\}$  with respect to  $\mathcal{K}_m(A, q_1)$ . Since  $u_1, \dots, u_k$  are the optimal approximations in the sense of residual minimizations with respect to the spectral norm and  $\tilde{\mu}_i, i = 1, 2, \dots, k$ , they are generally more accurate than  $\varphi_i, i = 1, 2, \dots, k$ . Then the deflated subspace  $\text{span}\{u_1, \dots, u_k\}^\perp$  contains more information on the unwanted eigenvectors than the deflated subspace  $\text{span}\{\varphi_1, \dots, \varphi_k\}^\perp$

does. As is well known, if a subspace contains more information on the desired eigenvectors, then the Ritz values with respect to it are generally more accurate [5, 9, 12]. So the Ritz values  $\xi_j$ ,  $j = 1, 2, \dots, m - k$  with respect to  $\text{span}\{u_1, \dots, u_k\}^\perp$  are usually more accurate approximations to some of the unwanted eigenvalues than those with respect to  $\text{span}\{\varphi_1, \dots, \varphi_k\}^\perp$ . As shown qualitatively in [7],  $\xi_j$ ,  $j = 1, 2, \dots, m - k$  are theoretically better shifts than the exact shifts  $\tilde{\mu}_j$ ,  $j = k + 1, \dots, m$ . So we take  $\xi_j$ ,  $j = 1, \dots, m - k$  as shifts for use within SHIRRA, called the refined shifts. For an efficient and reliable computation of the refined shifts, we refer to [7] for details.

#### 4. PRACTICAL IMPLEMENTATIONS

##### 4.1. Eigenvalue computation

Like SHIRA, SHIRRA handles the matrix  $A = R_1(\sigma, W)$  or  $R_2(\sigma, W)$  and computes its  $k$  eigenvalues  $\mu_i$ ,  $i = 1, \dots, k$  and the associated eigenvectors  $y_i$ 's. But what we need are the eigenvalues  $\lambda$ 's of the Hamiltonian matrix  $W$ . We modify the approach used in [11] slightly and describe it as follows.

If the approximate eigenvalues  $\tilde{\mu}_1, \dots, \tilde{\mu}_k$  and the refined eigenvector approximations  $u_1, \dots, u_k$  have converged, we get a converged invariant subspace  $\text{span}\{u_1, \dots, u_k\}$  of  $R_1(\sigma, W)$ . A feasible approach is to calculate the Ritz values of  $W^2$  with respect to the subspace  $\text{span}\{u_1, \dots, u_k\}$ ; that is, we compute the eigenvalues  $\gamma_i$  of  $B_k = V_k^T W^2 V_k$ , where  $V_k$  is an orthonormal basis of  $\text{span}\{u_1, \dots, u_k\}$ . Then  $\pm\sqrt{\gamma_i}$  are the eigenvalues of  $W$ . As a byproduct, we obtain  $W^2 V_k$  when forming  $B_k$ , which is used to compute the residual norm

$$\|W^2 V_k - V_k B_k\|_F,$$

and we check whether or not  $\text{span}\{u_1, \dots, u_k\}$  is approximately invariant under  $W^2$ . This test is necessary because it can happen that a space that is invariant under  $R_1(\sigma, W)$  fails to be invariant under  $W^2$ . The failure is rare and happens when  $\sigma$  is chosen such that two distinct eigenvalues of  $W^2$  are mapped to the same eigenvalue of  $A$ ; see [11] for arguments.

##### 4.2. Stopping criterion

In order to design a reliable and comparable stopping criterion for SHIRRA and SHIRA, we need to discuss the issue of how good the subspace spanned by the approximate eigenvectors is when individual approximate eigenpairs are accurate enough.

Let  $(\tilde{\mu}_i, \tilde{y}_i)$ ,  $1, 2, \dots, k$  be approximate eigenpairs, e.g., the Ritz pairs or refined Ritz pairs, of  $A$  that satisfy

$$(13) \quad \|A\tilde{y}_i - \tilde{\mu}_i\tilde{y}_i\| \leq \epsilon$$

with  $\epsilon$  a user-prescribed accuracy. Define the matrices

$$\tilde{D} = \text{diag}(\tilde{\mu}_1, \dots, \tilde{\mu}_k), \quad \tilde{Y} = (\tilde{y}_1, \dots, \tilde{y}_k).$$

Then from (13) we get

$$(14) \quad \|A\tilde{Y} - \tilde{Y}\tilde{D}\|_F \leq \sqrt{k} \epsilon.$$

Let  $\tilde{Y} = QR$  be the (thin) QR factorization of  $\tilde{Y}$ . Then the columns of  $Q$  and those of  $\tilde{Y}$  span the same subspace. From (14) and norm properties, we get the inequalities

$$(15) \quad \frac{\|A\tilde{Y} - \tilde{Y}\tilde{D}\|_F}{\|R\|} \leq \|AQ - QR\tilde{D}R^{-1}\|_F \leq \sqrt{k}\|R^{-1}\|_F \epsilon.$$

These relationships indicate that the residual norm of  $\text{span}\{Q\}$  is amplified by at most  $\sqrt{k}\|R^{-1}\|_F$  times, relative to those of individual approximate eigenpairs. In our algorithm, we use the residual norms of individual approximate eigenpairs other than the residual norm of the subspace to decide if the algorithm converges.

### 4.3. Eigenvector computation

Since the eigenvectors of  $W^2$  are generally not those of  $W$ , SHIRRA may fail to recover the desired eigenvectors of  $W$  from those of  $A$ . Mehrmann and Watkins [11] proposed to compute the desired eigenvectors of  $W$  by using inverse iteration after  $\lambda_i$ ,  $i = 1, 2, \dots, k$  are computed. It does not make use of the subspace information obtained by SHIRRA. Hwang, Lin and Mehrmann [4] present a more effective procedure to compute a stable invariant subspace of  $W$  directly from the invariant subspace of  $W^2$ . They have proved

$$(16) \quad \text{span}\{V^-\} = \text{span}\{WQ_k - Q_kX_k\},$$

where  $\text{span}\{V^-\}$  is the stable subspace of  $W$ ,  $\text{span}\{Q_k\}$  is the invariant subspace of  $W^2$  and  $X_k$  is the positive square root of the matrix  $\Omega_k = Q_k^T W^2 Q_k$  satisfying  $X_k^2 = \Omega_k$ . If we get the invariant subspace  $\text{span}\{Q_k\}$  of  $W^2$ , we can compute the invariant subspace  $\text{span}\{V^-\}$  of  $W$  directly. Then we can easily get the desired eigenvalues and eigenvectors. We have used this approach to determine a stable invariant subspace of  $W$  in SHIRRA.

## 5. NUMERICAL EXPERIMENTS

We now use SHIRRA to solve some QEPs

$$\lambda^2 Mx + \lambda Gx + Kx = 0$$

and compare it with SHIRA, IRRA and eigs. All the experiments were performed on an Intel(R) core(TM)2 with CPU 1.86GHz and 1GB RAM using Matlab 7.1 with  $\epsilon_{\text{mach}} = 2.22 \times 10^{-16}$  under the Window XP system.

**Example 1.** We consider a QEP arising in the vibration analysis of a wire-saw [2], where the  $n \times n$  coefficient matrices are defined by  $M = 2I_n$ ,  $K = \text{diag}_{1 \leq j \leq n}(j^2 \pi^2 (1 - \nu^2)/2)$  and

$$G = -G^* = (g_{ij}) \text{ with } g_{ij} = \begin{cases} \frac{4ij}{j^2 - i^2} \nu & \text{if } i + j \text{ is odd,} \\ 0 & \text{otherwise.} \end{cases}$$

Here  $\nu$  is a real nonnegative parameter and we took  $n = 3000$  and  $\nu = 1.1$ . We point out that  $G$  is almost dense, so that the resulting  $W$  is quite dense. As a consequence, the matrix-vector products involving  $(W - \sigma I)^{-1}$  and  $R_1(\sigma, W)$  or  $R_2(\sigma, W)$  are costly, and the isotropic Arnoldi process and the standard Arnoldi process are relatively expensive. Particularly, matrix-vector products dominate the computational cost of the two processes for not big steps. In this case, each step of the isotropic Arnoldi process is much more costly than that of the standard Arnoldi process since a matrix-vector product in the former amounts to four or two matrix-vector products in the latter, depending on use of  $R_1(\sigma, W)$  or  $R_2(\sigma, W)$ .

We computed the six eigenvalues that are closest to the complex target  $\sigma_1 = 1.2 - 115i$  and the purely imaginary target  $\sigma_2 = 22i$ . So for these two targets, we used  $R_1(\sigma, W)$  and  $R_2(\sigma, W)$ , respectively. We compared SHIRRA with SHIRA, eigs and IRRA, starting with the same initial vector generated randomly in a uniform distribution. We applied eigs and IRRA to  $(W - \sigma I)^{-1}$  in complex arithmetic. In experiments, we used the stopping tolerance  $10^{-9}$ , and the maximum number of restarts was limited to 20. The subspace size was 22 and we used 12 shifts per implicit restart. Table 1 reports the number of restarts and the CPU timings in second for eigenvalue computation.

Table 1: Example 1

targets $\sigma$	Restarts				CPU(s)			
	SHIRRA	SHIRA	IRRA	eigs	SHIRRA	SHIRA	IRRA	eigs
$1.2 - 115i$	2	6	6	5	165.2	354.6	207.5	174.6
$22i$	3	10	4	4	192.4	480.0	83.6	83.7

For SHIRRA and SHIRA, we only computed the desired eigenvalues. When the associated eigenvectors or invariant subspace is required, we used the stable subspace method [4] to compute. For example, the CPU time of SHIRRA for eigenvector computation was 20.03s for  $\sigma = 1.2 - 115i$ . It is much less than the CPU time of SHIRRA for computing the desired eigenvalues.



Some observations on Table 1 are in order. First, the four algorithms worked quite well for this problem and they found the desired eigenvalues quickly. Second, regarding restarts, SHIRRA are three times as fast as SHIRA; regarding CPU timings, SHIRRA was more than twice as fast as SHIRA. Third, since  $W$  is quite dense, matrix-vector products heavily dominate the computational costs in the isotropic Arnoldi process and the standard Arnoldi process for not big Arnoldi steps. Furthermore, the average CPU time of SHIRRA and SHIRA per restart was considerably more than that of IRRA and eigs, and SHIRRA and SHIRA cost more than IRRA and eigs. There are two reasons for them. The first reason is that SHIRRA and SHIRA was coded in Matlab, while the Arnoldi process in IRRA and eigs call the much more efficient Fortran code ARPACK. The second reason is that a matrix-vector product in SHIRRA and SHIRA is considerably more costly than that in IRRA and eigs, as commented above.

Next we check the approximate subspace residual norm  $\|W^2V_k - V_kB_k\|_F$  of  $W^2$  for the convergence of SHIRRA and SHIRA and the tolerance was  $10^{-6}$ . We depict  $\|W^2V_k - V_kB_k\|_F$  versus restarts for  $k = 6$  in Figure 1, where the left figure is the convergence curve for  $\sigma = 1.2 - 115i$  and the right figure is for  $\sigma = 22i$ . For both targets, it is clearly seen that  $\|W^2V_k - V_kB_k\|_F$  obtained by SHIRRA tended to zero very quickly and smoothly, while that obtained by SHIRA decreased in the first very few restarts and then stagnated at the level of  $O(10^{-2})$  for all restarts. Therefore, SHIRRA is much more effective than SHIRA for computation of the invariant subspace.

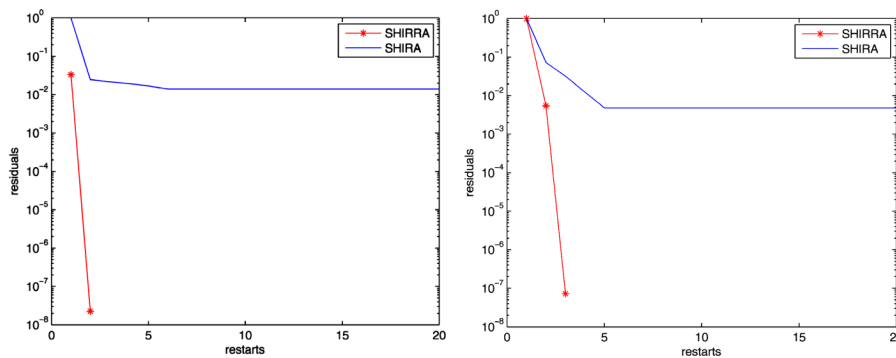


Fig. 1. Left:  $\sigma = 1.2 - 115i$ ; Right:  $\sigma = 22i$ .

Finally, we report the results obtained by SHIRA with increased subspace sizes 27, 32 and numerous shifts and compare it with SHIRRA with smaller subspace size 22; see Table 2, where ‘Shifts number’ was the number of shifts used in each restart. Since the number of shifts was variable, the dimension  $k$  of the invariant subspace computed by SHIRA might not equal to the number of the wanted eigenvalues, we listed the residual norms  $\|W^2V_k - V_kB_k\|_F$ , denoted by ‘Residual’, of  $k$ -dimensional approximate invariant subspaces.

Table 2: Example 1 for SHIRA

targets	Shifts number	Subspace size	Restarts	CPU(s)	Residual	$k$
$\sigma = 1.2 - 115i$	12	27	6	320.23	$5.13 \times 10^{-9}$	7
$\sigma = 1.2 - 115i$	12	32	6	333.46	$5.22 \times 10^{-9}$	7
$\sigma = 1.2 - 115i$	12	37	5	299.17	$5.52 \times 10^{-9}$	7
$\sigma = 1.2 - 115i$	16	27	5	322.58	$5.19 \times 10^{-9}$	11
$\sigma = 1.2 - 115i$	20	27	3	249.45	$2.43 \times 10^{-9}$	7
$\sigma = 22i$	12	27	10	442.28	$2.23 \times 10^{-7}$	6
$\sigma = 22i$	12	32	10	459.83	$7.57 \times 10^{-7}$	6
$\sigma = 22i$	12	37	8	424.05	$2.65 \times 10^{-8}$	7
$\sigma = 22i$	16	27	7	400.25	$4.45 \times 10^{-8}$	6
$\sigma = 22i$	20	27	4	301.30	$2.80 \times 10^{-8}$	7

We see from Table 2 that compared with SHIRA with subspace size 22 in Table 1, for two targets  $\sigma$ 's and increased subspace sizes 27 and 32, SHIRA could reduce restarts but in most cases it was considerably inferior to SHIRRA with smaller subspace size 22. These results demonstrate that choosing a better refined method and better refined shifts was more important than increasing subspace size and changing number of shifts in SHIRA.

**Example 2.** In this example, we take the matrices from the butterfly problem as described in [2]. This is a quadratic matrix polynomial of dimension  $m^2$ . Its coefficient matrices are Kronecker products with  $A_4$  and  $A_2$  real and symmetric and  $A_3$  and  $A_1$  real and skew-symmetric. For our test purpose, we set  $M = A_4$ ,  $G = A_3$  and  $K = A_2$  to obtain a QEP considered in the current paper.

In experiments, we took  $m = 100$  and obtained a  $10000 \times 10000$  QEP. Moreover, we set the involved parameter vector  $c = [0.1, 1.2, 1.0, 1.0, 1.2, 1.0]$ , the same as the original one in [2]. We are interested in the six eigenvalues that are closest to the complex targets  $\sigma_1 = -0.9 - 5.5i$  and  $\sigma_2 = 0.8 + 4.8i$ .

We used the four algorithms to find the desired eigenvalues. The maximum number of restarts was 50 for SHIRRA and SHIRA and was 100 for eigs and IRRA. The subspace size in each restart was 22 and we used 10 shifts per implicit restart. The stopping tolerance was  $10^{-9}$ . We also checked the subspace residual  $\|W^2 V_k - V_k B_k\|_F$  of  $W^2$  for the convergence of SHIRRA and SHIRA and the tolerance was  $10^{-6}$ . Some results were listed in Table 3 for eigenvalue computation.

Table 3: Example 2

targets $\sigma$	Restarts				CPU(s)			
	SHIRRA	SHIRA	IRRA	eigs	SHIRRA	SHIRA	IRRA	eigs
$-0.9 - 5.5i$	6	20	-	-	53.5	162.6	-	-
$0.8 + 4.8i$	7	10	-	-	57.4	93.9	-	-

The results indicate that IRRA and eigs failed to converge for the two targets after 100 restarts were used. In contrast, SHIRRA and SHIRA solved the problem successfully. SHIRRA outperformed SHIRA very considerably. In terms of restarts and CPU timings, the former was three times as fast as the latter for  $\sigma = -0.9 - 5.5i$  and one and a half times as fast as SHIRA for  $\sigma = 0.8 + 4.8i$ .

We depict the curves of  $\|W^2V_k - V_kB_k\|_F$  versus restarts for the given two targets in Figure 2. We see that  $\|W^2V_k - V_kB_k\|_F$  obtained by SHIRRA dropped below the prescribed tolerance  $10^{-6}$ . However, SHIRA failed to compute the invariant subspaces for the two targets. For  $\sigma = -0.9 - 5.5i$ ,  $\|W^2V_k - V_kB_k\|_F$  decreased slowly and smoothly in the first 20 restarts and then stagnated around  $10^{-4}$  in later 30 restarts. For  $\sigma = 0.8 + 4.8i$ , SHIRA behaved worse.  $\|W^2V_k - V_kB_k\|_F$  experienced three stages. In the first 5 restarts, it decreased quite slowly. Then in the five restarts followed, it was irregular but did not exhibit significant reduction. From the 11th to the last restart, it stagnated completely and only reached  $10^{-2}$ , much bigger than  $10^{-6}$ . Again, this example indicates that SHIRRA is much better than SHIRA for computing the desired invariant subspace.

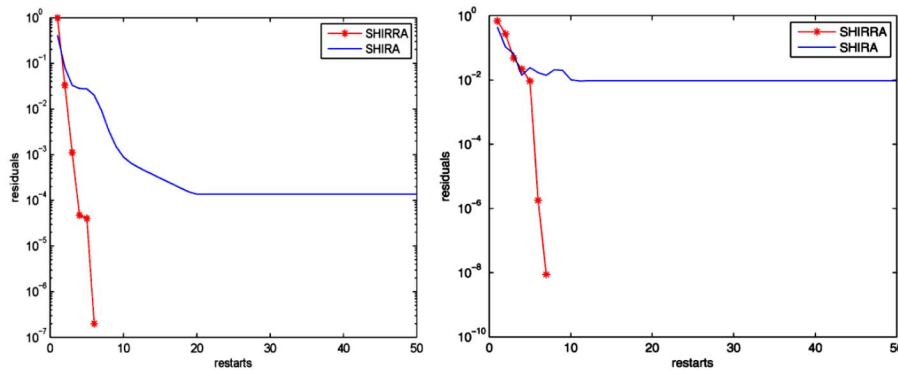


Fig. 2. Left:  $\sigma = -0.9 - 5.5i$ ; Right:  $\sigma = 0.8 + 4.8i$ .

We see from Table 4 that for two targets  $\sigma$ 's and increased subspace sizes 27 and 32, SHIRA behaved irregularly and did not necessarily use fewer restarts. In most cases, SHIRRA with smaller subspace size 22 was much more efficient than SHIRA with subspace sizes 27 and 32. Still, these results indicate that it was the better method and better shifts that play a more important role than simply increasing subspace size in SHIRA.

**Example 3.** This problem comes from [11]. We construct some matrices of order  $n = m^2$  by a tensor product construction. Let  $B$  denote the  $m \times m$  nilpotent Jordan block

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

Finally, we report the results obtained by SHIRA with increased subspace size and compare it with SHIRRA with smaller subspace size 22. Table 4 lists the results.

Table 4: Example 2 for SHIRA

targets	Shifts number	Subspace size	Restarts	CPU(s)	Residual	$k$
$\sigma = -0.9 - 5.5i$	12	27	24	190.49	$7.00 \times 10^{-14}$	15
$\sigma = -0.9 - 5.5i$	12	32	29	253.71	$7.00 \times 10^{-14}$	13
$\sigma = -0.9 - 5.5i$	12	37	31	259.77	$7.00 \times 10^{-14}$	13
$\sigma = -0.9 - 5.5i$	16	27	6	69.08	$1.07 \times 10^{-10}$	11
$\sigma = -0.9 - 5.5i$	20	27	9	110.23	$1.00 \times 10^{-10}$	7
$\sigma = 0.8 + 4.8i$	12	27	15	128.92	$9.42 \times 10^{-10}$	10
$\sigma = 0.8 + 4.8i$	12	32	18	150.09	$4.71 \times 10^{-10}$	10
$\sigma = 0.8 + 4.8i$	12	37	22	189.22	$1.27 \times 10^{-9}$	10
$\sigma = 0.8 + 4.8i$	16	27	7	78.86	$8.96 \times 10^{-12}$	10
$\sigma = 0.8 + 4.8i$	20	27	6	82.09	$2.56 \times 10^{-9}$	7

and define

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

Then we set

$$(17) \quad \begin{aligned} M &= c_{11}I_m \otimes \tilde{M} + c_{12}\tilde{M} \otimes I_m, \\ G &= c_{21}I_m \otimes \tilde{G} + c_{22}\tilde{G} \otimes I_m, \\ K &= c_{31}I_m \otimes \tilde{K} + c_{32}\tilde{K} \otimes I_m, \end{aligned}$$

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

Taking  $m = 150$  and the coefficients

$$\begin{aligned} c_{11} &= 2, & c_{12} &= 2, \\ c_{21} &= 1.5, & c_{22} &= 11, \\ c_{31} &= 21.4, & c_{32} &= 2.5, \end{aligned}$$

we get a  $W$  of order 45000. Set  $\sigma = -0.75 - 4.5i$ . We applied SHIRRA and SHIRA to  $R_1(\sigma, W)$  as well as eigs and IRRA to  $(W - \sigma I)^{-1}$  in complex arithmetic. We used subspace size 22 and 10 shifts per implicit restart in the four algorithms and the stopping tolerance was set to  $10^{-9}$ . Here we used SHIRRA and SHIRA to compute the six eigenvalues closest to  $\sigma$ , which are three complex conjugate pairs and correspond to

three quadruplets of the eigenvalues of (1). We then got the other six eigenvalues closest to the target by taking minus of theirs. eigs and IRRA were applied to  $(W - \sigma I)^{-1}$  directly for finding the six eigenvalues that are closest to the target. We then took their complex conjugates to get the other six ones. Again, we used ten Arnoldi steps per implicit restart. We listed the number of restarts and the CPU timings for eigenvalue computation in Table 5.

Table 5: Example 3

	SHIRRA	SHIRA	IRRA	eigs
Restarts	19	32	20	35
CPU(s)	434.73	658.80	97.06	174.95

It is observed from the table that the four algorithms worked well but SHIRRA outperforms SHIRA by a factor of two to three in terms of both restarts and CPU timings. Both SHIRRA and SHIRA were more efficient than IRRA and eigs, respectively. Meanwhile, we see that IRRA was considerably better than eigs. So SHIRRA is much more efficient than SHIRA, IRRA and eigs.

For the computation of the desired invariant subspace, just as done in Examples 1–2, we depict the curve of  $\|W^2V_k - V_kB_k\|_F$  versus restarts in Figure 3 for SHIRA and SHIRRA with subspace 22. Unlike Examples 1-2, for this example we see from the figure that both SHIRRA and SHIRA found the desired invariant subspace successfully, but SHIRRA was considerably faster than SHIRA.

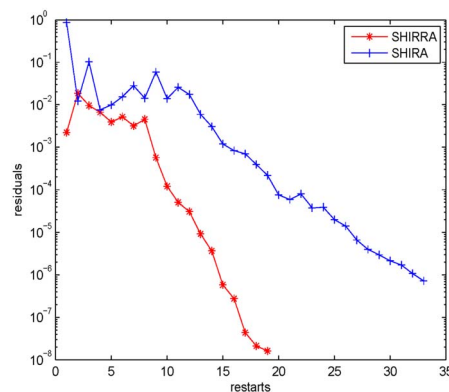


Fig. 3. Example 3.

Finally, we report the results obtained by SHIRA with much increased subspace sizes 25, 30 and 35 and compare it with SHIRRA with much smaller subspace size 10. Table 6 lists the results. It is observed from Table 6 that for numerous shifts and increased subspace sizes 25, 30 and 35, SHIRA behaved irregularly and did not

necessarily use fewer restarts. In all cases, SHIRRA with smaller subspace size 10 was much more efficient than SHIRA with dimension sizes 25, 30 and 35. Again, these results indicate that it was the better method and better shifts that play a major role and solved the problem more effectively.

Table 6: Example 3 for SHIRA

target	Shifts number	Subspace size	Restarts	CPU(s)	Residual	$k$
$\sigma = -0.75 - 4.5i$	10	25	33	666.31	$7.26 \times 10^{-7}$	6
$\sigma = -0.75 - 4.5i$	10	30	27	577.06	$7.40 \times 10^{-7}$	6
$\sigma = -0.75 - 4.5i$	10	35	30	660.92	$7.22 \times 10^{-7}$	6
$\sigma = -0.75 - 4.5i$	14	25	23	645.73	$4.65 \times 10^{-7}$	6
$\sigma = -0.75 - 4.5i$	18	25	39	1354.28	$1.05 \times 10^{-7}$	6

**Example 4.** We consider the QEP obtained by a finite element discretization of equations of elastic deformation of an anisotropic material [11]. The matrix has order 2223.

We computed the twelve eigenvalues closest to the imaginary axis. The six smallest eigenvalues in the right half-plane was given in [11]. They are a priori known to lie near the real axis, so it is reasonable to use real targets. We used ten Arnoldi steps per implicit restart and the stopping tolerance  $10^{-10}$ . Table 7 lists the results. We saw that SHIRRA and SHIRA converged very fast for this matrix but the former was faster than SHIRA by at least one restart. We had to compute all twelve eigenvalues when we used unstructured algorithms eigs and IRRA for  $\sigma = 0$ . For  $\sigma = 0.1$ , it is closer to  $-\lambda_1$  than  $\lambda_6$ , so we had to compute nine eigenvalues in order to find  $\lambda_6$ . If no priori information is known, we can reasonably select a purely imaginary target. SHIRRA and SHIRA worked excellently for the purely imaginary target  $\sigma = 0.1i$ , but eigs and IRRA did not deliver the correct eigenvalues. For this problem, both SHIRRA and SHIRA were superior to IRRA and eigs considerably, while IRRA and eigs used the same restarts and comparable CPU timings.

Table 7: Example 4

targets $\sigma$	Restarts				CPU(s)			
	SHIRRA	SHIRA	IRRA	eigs	SHIRRA	SHIRA	IRRA	eigs
0	3	4	7	7	1.33	1.36	1.85	1.67
0.1	3	4	10	10	1.38	1.63	1.86	1.72
0.6	2	4	4	4	0.98	1.56	1.03	0.98
$0.1i$	3	4	-	-	1.38	1.76	-	-

## 6. CONCLUSION

We have considered numerical solution of SHH pencil eigenvalue problems that stems from linearizations of certain QEPs. Combining SHIRA with the refined pro-

jection principle, we have developed a refined variant of SHIRA algorithm (SHIRRA) for effectively computing a few eigenvalues and the associated eigenvectors of a large SHH pencil. Based on the refined eigenvector approximations, we have proposed refined shifts for use within SHIRRA that are theoretically better and numerically more efficient than the exact shifts for use within SHIRA. Numerical experiments have illustrated that SHIRRA can outperform SHIRA considerably and often speed up IRRA very much. Also, numerical experiments have illustrated that SHIRRA is superior to SHIRA for eigenvector (invariant subspace) computation.

The Matlab code of SHIRRA is available and can be obtained from us.

#### ACKNOWLEDGMENTS

We thank Professor David Watkins and Professor TsungMin Hwang for providing us their Matlab codes.

#### REFERENCES

1. P. Benner, H. Faßbender and M. Stoll, A Hamiltonian Krylov-Schur-type method based on the symplectic Lanczos process, *Linear Algebra Appl.*, (2010), doi:10.1016/j.laa.2010.04.
2. T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, Technical Report 2008.40, MIMS, University of Manchester, Apr., 2008.
3. G. H. Golub and C. F. Van Loan, *Matrix Computations*, The 3rd Edition, The John Hopkins University, Baltimore, 1996.
4. T.-M. Hwang, W.-W. Lin and V. Mehrmann, Numerical solution of quadratic eigenvalue problems with structure-preserving methods, *SIAM J. Sci. Comput.*, **24** (2003), 1283-1302.
5. Z. Jia, The convergence of generalized Lanczos methods for large unsymmetric eigenproblems, *SIAM J. Matrix Anal. Appl.*, **16** (1995), 843-862.
6. Z. Jia, Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, *Linear Algebra Appl.*, **259** (1997), 1-23.
7. Z. Jia, Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and implicitly restarted refined Arnoldi algorithm, *Linear Algebra Appl.*, **287** (1999), 191-214.
8. Z. Jia, A refined subspace iteration algorithm for large sparse eigenproblems, *Appl. Numer. Math.*, **32** (2000), 35-52.
9. Z. Jia and G. W. Stewart, An analysis of the Rayleigh-Ritz method for approximating eigenspaces, *Math. Comput.*, **70** (2001), 637-647.

10. Z. Jia, Some theoretical comparisons of refined Ritz vectors and Ritz vectors, *Sci. China, Ser. A*, **47** (Suppl.), (2004), 222-233.
11. V. Mehrmann and D. Watkins, Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils, *SIAM J. Sci. Comput.*, **22** (2001), 1905-1925.
12. Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
13. D. C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Appl.*, **13** (1992), 357-385.
14. G. W. Stewart, *Matrix Algorithms II: Eigensystems*, SIAM, Philadelphia, PA, 2001.
15. H. A. Van der Vorst, *Computational Methods for Large Eigenvalue Problems*, Elsevier, North-Hollands, 2002.
16. C. F. Van Loan, A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix, *Linear Algebra Appl.*, **61** (1984), 233-251.

Zhongxiao Jia  
Department of Mathematical Sciences  
Tsinghua University  
Beijing 100084  
People's Republic of China  
E-mail: jiazx@tsinghua.edu.cn

Yuquan Sun  
LMIB & School of Mathematics and Systems Science  
BeiHang University  
Beijing 100191  
People's Republic of China  
E-mail: sunyq@buaa.edu.cn