Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm

Zhongxiao Jia *

Department of Applied Mathematics, Dalian University of Technology, Dalian 116024,
People's Republic of China

Received 13 November 1997; accepted 21 September 1998

Submitted by V. Mehrmann

Dedicated to Professor Ludwig Elsner on his 60th birthday, whose insight and style in research have been affecting the author’s academic career

Abstract

It is well known that a Ritz vector obtained by the Lanczos method and the Arnoldi method can be characterized by an elegant explicit polynomial, whose roots are just the other $m - 1$ Ritz values of the matrix $A$ in question from the $m$-dimensional Krylov subspace involved. Analogous results hold for the recently developed harmonic Lanczos and Arnoldi methods, and the roots of the corresponding polynomial are now the other $m - 1$ harmonic Ritz values of $A$ from the same subspace. In this paper, we investigate a polynomial characterization of the refined Arnoldi method proposed by the author in recent years. We derive a polynomial characterization of the refined Arnoldi method and apply the implicitly restarting scheme proposed by Sorensen to the refined Arnoldi method. The roots of this polynomial are used as shifts, called refined shifts, within an implicitly restarted refined Arnoldi algorithm. The shifts are interpreted to have the

* Corresponding author. E-mail: zxjia@gingko.dlut.edu.cn.

1 Supported by the China State Key Project for Basic Researches, the National Natural Science Foundation of China, the Doctoral Program of the China State Educational Commission, the Foundation of Chinese Returned Scholars and the Natural Science Foundation of Liaoning Province, China.
same nature as the exact shifts used within the implicitly restarted Arnoldi algorithm proposed by Sorensen. Numerical experiments compare the use of refined shifts with exact shifts within an implicitly restarted Arnoldi algorithm. The results also show that implicitly restarting the refined Arnoldi method is far superior to the Saad's explicitly restarted scheme. © 1999 Elsevier Science Inc. All rights reserved.

1. Introduction

The Lanczos method [17] and the Arnoldi method [1] are among the most commonly used methods for large-scale symmetric (Hermitian) and nonsymmetric (non-Hermitian) eigenproblems, see, e.g. [6,27,28]. Both methods are orthogonal projection methods, and they work as follows: Given an \( m \)-dimensional Krylov subspace \( \mathcal{K}_m(v_1,A) \) generated by a starting vector \( v_1 \) and the matrix \( A \), they use the \( m \) Ritz pairs of \( A \) from \( \mathcal{K}_m(v_1,A) \) to approximate \( m \) eigenpairs of \( A \). For detailed theoretical analyses, we refer the reader to [6,27] and [7,8,11,12,28], respectively.

The harmonic Lanczos and harmonic Arnoldi methods are two important methods proposed and developed recently by a few researchers in Refs. [5,21–24,26], and they use harmonic Ritz pairs to approximate \( m \) eigenpairs of \( A \). The harmonic Ritz values are the reciprocals of the Ritz values of \( A^{-1} \) from the subspace \( A\mathcal{K}_m(v_1,A) \), and accordingly the harmonic Ritz pairs satisfy the so-called harmonic Rayleigh–Ritz projection. We refer to the above-mentioned references and [29] for details and extensions to other methods, such as the Jacobi–Davidson method.

Of important results on the Lanczos and Arnoldi methods as well as on their harmonic versions, a very attractive one is that both a Ritz vector and a harmonic Ritz vector can be characterized by certain elegant \( m-1 \)-degree explicit polynomials, whose roots are the other \( m-1 \) Ritz values and other \( m-1 \) harmonic Ritz values, respectively, e.g., see [24,27] for the proof. Polynomial characterizations of some quantities obtained by Krylov subspace type methods, such as Ritz vectors, harmonic Ritz vectors, residual vectors, etc., are of major theoretical interest both for eigenproblems and linear systems. There has been considerable research on this important subject, e.g., [5,21,24,26,27,31]. The polynomial characterizations are not intended to be used for computational purposes, instead they attempt to lead to better understanding of these quantities and of the methods and to expose natures of them, as is also seen from the above-mentioned references.

To overcome possible non-convergence of Ritz vectors and of the Arnoldi method [7,8,11,16], a refined strategy has been proposed in [7,9], and it has been extended to other orthogonal projection methods, e.g., the block Arnoldi method [10] and subspace iteration [14]. The key idea is that the refined methods only retain Ritz or harmonic Ritz values but instead of computing
Ritz or harmonic Ritz vectors they seek a refined Ritz vector for each Ritz or harmonic Ritz value that minimizes norm of the residual formed with the Ritz or harmonic Ritz value over the subspace involved.

A systematic and rigorous convergence theory for the refined orthogonal projection methods has been attempted in [13,15,16], and a number of important results have been derived. Compared to the theory of the Lanczos and Arnoldi methods, among other things, however, one of the important problems unsolved is that we did not derive elegant polynomial characterizations of refined Ritz vectors. How to carry out this significant task is one of the two main goals of this paper. The other goal of the paper is to improve the refined Arnoldi algorithm proposed in [9], where the explicitly restarting scheme of Saad [28], p. 234, was adopted. What we will do is to apply the implicitly restarting scheme proposed by Sorensen [30] to the refined Arnoldi method and to find new shifts. The implicitly restarting scheme is directly applicable, and to this point our contribution is to find new shifts based on the refined Ritz vectors for use within an implicitly restarted refined Arnoldi algorithm. Under certain decompositions of orthogonal direct sums, the new shifts, called the refined shifts, are shown to have the same nature as the exact shifts proposed by Sorensen for use within an implicitly restarted Arnoldi algorithm [30]. In other words, in the framework of an implicitly restarted refined Arnoldi algorithm, the refined shifts are interpreted to be nothing but the counterparts of the exact shifts proposed by Sorensen for use within an implicitly restarted Arnoldi algorithm [30]. We show that implicitly restarting using either exact or refined shifts on the convection–diffusion problem is dramatically more efficient than the restarting scheme in [9] adapted from Saad [28], p. 234. In the meanwhile, we mainly draw comparisons for our algorithm and Sorensen's algorithm on four problems.

Although, as indicated in [18–20,30], any set of shifts can be used within an implicitly restarted Arnoldi algorithm, there are two ingredients to an implicitly restarted refined Arnoldi algorithm. The first is using the roots of the polynomial characterizing a refined Ritz vector as shifts. The second is approximating a wanted eigenvector by a refined (harmonic) Ritz vector instead by a (harmonic) Ritz vector.

In Section 2 we briefly review the Arnoldi method, the harmonic Arnoldi method and the refined Arnoldi method. In Section 3 we derive an explicit polynomial characterization of each refined Ritz vector. It is proved that the roots of the polynomial are the \( m - 1 \) Ritz values of \( A \) from the orthogonal complement of span of the refined approximate eigenvector with respect to \( \mathcal{N}_m(v_1,A) \). These roots are more accurate than those other \( m - 1 \) Ritz values. In Section 4 we briefly review the implicitly restarted Arnoldi algorithm proposed by Sorensen [30]. In Section 5 we apply the implicitly restarted scheme to the refined Arnoldi method and make use of the arguments and observations in the paper to propose the refined shifts based on the refined Ritz vectors for use within an implicitly restarted refined Arnoldi algorithm. The refined shifts are
theoretically better than the exact shifts used within the implicitly restarted Arnoldi method proposed by Sorensen [30]. Finally, we report numerical experiments to illustrate that the resulting implicitly restarted refined Arnoldi algorithm and the algorithm of Sorensen are dramatically more efficient than the restarting scheme used in [9]. Comparisons are mainly drawn for our algorithm and that of Sorensen.

Let us introduce some notations to be used. Denote by the superscript * the conjugate transpose of a matrix or a vector, by e_i the ith coordinate vector of dimension i, and by ∥x∥ = (x,x)^{1/2} the usual Euclidean norm. We denote by $P_n$ the set of all complex polynomials with degree at most n and by $\sigma_{\min}(X)$ the smallest singular value of X.

2. The Arnoldi method, its harmonic and refined versions

Consider the matrix eigenproblem

$$A\varphi_i = \lambda_i \varphi_i,$$

where $A \in \mathbb{C}^{N\times N}$ is a large diagonalizable matrix, and $\lambda_i, \varphi_i, i = 1, 2, \ldots, N$ are its eigenpairs with $\|\varphi_i\| = 1$.

Given the Krylov subspace $\mathcal{K}_m(v_1,A)$ generated by a unit-length starting vector $v_1$ and $A$, assume that $\dim(\mathcal{K}_m(v_1,A)) = m$. Then the Arnoldi process [28] generates an orthonormal (unitary) basis $\{v_k\}_{k=1}^m$, and it can be written in the matrix form

$$AV_m = V_mH_m + h_{m+1}v_{m+1}e_m^* = V_{m+1}\tilde{H}_m,$$

where $V_k = (v_1, v_2, \ldots, v_k)$ for $k = m, m+1$ and $H_m, \tilde{H}_m$ are the $m \times m$ and $(m + 1) \times m$ Hessenberg matrices with the entries $h_{ij}$, respectively.

The Arnoldi method is a Rayleigh–Ritz procedure or more generally a Galerkin projection [28]. It computes $m$ Ritz pairs $\mu_i, \tilde{\varphi}_i$ from $\mathcal{K}_m(v_1,A)$ as follows:

$$H_m x_i = \mu_i x_i,$$
$$\tilde{\varphi}_i = V_m x_i.$$  

The harmonic Arnoldi method is a harmonic Rayleigh–Ritz procedure. It seeks $m$ harmonic Ritz pairs $\theta_i, g_i$ from $\mathcal{K}_m(v_1,A)$ as follows [22–26]:

$$\tilde{H}_m^* \tilde{H}_m y_i = \theta_i H_m^* y_i,$$
$$g_i = V_m y_i.$$  

The refined Arnoldi method retains Ritz or harmonic Ritz values and seeks for each $\mu_i$ or $\theta_i$, represented as one symbol $\tilde{\lambda}_i$ from now on, a unit norm vector $u_i \in \mathcal{K}_m(v_1,A)$ that satisfies the following optimality property [7,9,13]
\[(A - \tilde{\lambda}_iI)u_i = \min_{u \in \mathcal{N}(v_1, A), \|u\| = 1} \|(A - \tilde{\lambda}_iI)u\|, \quad (2.5)\]

and uses it to approximate \(\phi_i\). So \(u_i\) is the best approximation to \(\phi_i\) from \(\mathcal{N}(v_1, A)\) with respect to \(\tilde{\lambda}_i\) and the Euclidean norm. Let \(u_i\) be a refined Ritz vector. Here we comment that \(\tilde{\lambda}_i\) can be any other approximate eigenvalue available, e.g., Rayleigh quotient of the harmonic Ritz vector.

Assume \(z_i\) to be the right singular vector associated with the smallest singular value \(\sigma_{\min}(H_m - \tilde{\lambda}_iI)\) of the \((m + 1) \times m\) Hessenberg matrix \(H_m - \tilde{\lambda}_iI\) where \(I\) is the same as the identity matrix with an additional zero row. Then it is readily shown \([7,9]\) that

\[\|(A - \tilde{\lambda}_iI)u_i\| = \sigma_{\min}(H_m - \tilde{\lambda}_iI), \quad (2.6)\]

\[u_i = V_mz_i.\]

Here we should point out that a refined Ritz vector \(u_i\) no longer satisfies a Galerkin projection over \(\mathcal{N}(v_1, A)\), namely, \(Au_i - \tilde{\lambda}_iu_i\) is not orthogonal to \(\mathcal{N}(v_1, A)\). Also, we may use the Rayleigh quotient \(\rho_i = u_i^*Au_i = z_i^*H_mz_i\) to replace \(\tilde{\lambda}_i\) as an approximation to \(\lambda_i\). \(\rho_i\) may be more accurate than \(\tilde{\lambda}_i\) because it is easily shown that \(\|(A - \rho_iI)u_i\| \leq \|(A - \tilde{\lambda}_iI)u_i\|\).

It is proved in [15] that \(u_i \neq \tilde{\phi}_i\) unless \(\|(A - \lambda_iI)(\tilde{\phi}_i)\| = 0\), i.e., the Ritz pair is an exact eigenpair of \(A\). Furthermore, it is shown that if \(\|(A - \tilde{\lambda}_iI)(\tilde{\phi}_i)\| \neq 0\) then

\[\|(A - \tilde{\lambda}_iI)u_i\| \leq \|(A - \tilde{\lambda}_iI)(\tilde{\phi}_i)\|, \quad (2.7)\]

and if at least one other \(\tilde{\lambda}_j(\neq \tilde{\lambda}_i)\) is very ill conditioned then

\[\|(A - \tilde{\lambda}_iI)u_i\| \ll \|(A - \tilde{\lambda}_iI)(\tilde{\phi}_i)\| \quad (2.8)\]

may occur. Therefore, \(u_i\) may be in general (much) more accurate than \(\tilde{\phi}_i\). However, we should point out that (2.7) and (2.8) do not imply that \(u_i\) must be more accurate than the associated Ritz vector because accuracy also depends upon the conditioning of the associated invariant subspace that \(u_i\) is attempting to approximate. Similar conclusions may be trivially drawn from the results in [15] for the refined Arnoldi method and the harmonic Arnoldi method. Therefore, the refined Arnoldi method delivers smaller residuals and more accurate approximations to eigenvectors, compared with the other two methods.

We comment that the idea of minimizing over a subspace is not new \([25,32]\) and there is no theoretical support to show how one can benefit from this idea in these two references.

---

\[2\] The author thanks Rich Lehoucq for drawing his attention to these two references.
3. Main results and proofs

To compare with our later results, we first review the established results for Ritz and harmonic Ritz vectors [24,27].

**Theorem 3.1.** Assume that all the Ritz or harmonic Ritz values are distinct, and let $\phi_i$ be a Ritz or harmonic Ritz vector of $A$ associated with the Ritz or harmonic Ritz value $\tilde{\lambda}_i$ from $\mathcal{H}_m(v_1,A)$ and define $\phi_i = p(A)v_1$. Then

$$p(\lambda) = \alpha \prod_{j \neq i} (\lambda - \tilde{\lambda}_j), \quad (3.9)$$

where $\alpha$ is a normalizing factor, that is, the polynomial roots of $p(\lambda) = 0$ are the other $m-1$ Ritz or harmonic Ritz values of $A$ from $\mathcal{H}_m(v_1,A)$.

We now present one of the main results of this paper. For simplicity, we will drop subscript $i$ of some related quantities whenever making no ambiguity.

**Theorem 3.2.** Assume $\dim(\mathcal{H}_m(v_1,A)) = m$, and let $\mathcal{H}_m(v_1,A) = \text{span}\{u\} \oplus \text{span}\{u\}^\perp$ be the orthogonal direct sum of $\mathcal{H}_m(v_1,A)$ with $\text{span}\{u\}^\perp$ being the orthogonal complement of $\text{span}\{u\}$ with respect to $\mathcal{H}_m(v_1,A)$. Then there exists a vector $q_1 \in \text{span}\{u\}^\perp$ such that

$$u = p(A)q_1 \quad (3.10)$$

with

$$p(\lambda) = \alpha \prod_{j=1}^{m-1} (\lambda - \eta_j), \quad (3.11)$$

where the $\eta_j$'s are the Ritz values of $A$ from $\text{span}\{u\}^\perp \subset \mathcal{H}_m(v_1,A)$ and $\alpha$ is a normalizing factor. Furthermore, $\mathcal{H}_m(q_1,A) = \mathcal{H}_m(v_1,A)$ and $\text{span}\{u\}^\perp = \mathcal{H}_{m-1}(q_1,A)$.

**Proof.** For the $m \times m$ unreduced upper Hessenberg matrix $H_m = V_m^*AV_m$ in (2.2), suppose $\tilde{Z}_m = (\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_m)$ with $\tilde{z}_m = z$ is an orthogonal matrix such that

$$\tilde{Z}_m^*H_m\tilde{Z}_m = C_m \quad (3.12)$$

is an $m \times m$ upper Hessenberg matrix with nonnegative subdiagonal entries. Because $H_m$ is unreduced, the subdiagonal entries of $C_m$ are positive. In terms of a reverse order analogue of the implicit $Q$ theorem of [6], p. 367, given $\tilde{z}_m = z, \tilde{z}_1$ and $C_m$ is uniquely determined by the Arnoldi process applied to the matrix $H_m$.

Recall from (2.6) that $u = V_mz$. Define

$$Q_m = V_m\tilde{Z}_m = (q_1, \ldots, q_{m-1}, u),$$
and $Q_{m-1}$ to be the matrix consisting of the first $m-1$ columns of $Q_m$. Then (3.12) is transformed into

$$Q_m^*AQ_m = C_m.$$  \hspace{1cm} (3.13)

So $C_m$ is the Hessenberg matrix generated by the Arnoldi process applied to the matrix $A$ using the starting vector $q_1$ and the columns of $Q_m$ form an orthonormal basis of $\mathcal{H}_m(q_1, A)$.

Furthermore, $q_1$ is uniquely determined and $\mathcal{H}_m(q_1, A) = \mathcal{H}_m(v_1, A)$ because $\bar{z}_1$ is unique and $\text{span}\{Q_m\} = \text{span}\{V_mZ_m\} = \text{span}\{V_m\}$.

We have from (3.13)

$$AQ_{m-1} = Q_{m-1}C_{m-1} + c_{mm-1}w_{m-1}^*,$$

where $C_{m-1}$ is the $(m-1) \times (m-1)$ leading principal submatrix of $C_m$. Based on [24] or [28], p. 186, we have

$$u = p(A)q_1$$ \hspace{1cm} (3.14)

with $p(\lambda) \in \mathcal{P}_{m-1}$, where the roots $\eta_j, j = 1, 2, \ldots, m-1$ of $p(\lambda) = 0$ are the eigenvalues of $C_{m-1}$. These $\eta_j$'s are the Ritz values of $A$ from $\mathcal{H}_{m-1}(q_1, A) = \text{span}\{Q_{m-1}\} \subset \mathcal{H}_m(v_1, A)$.

Since

$$u = V_mz \perp \text{span}\{Q_{m-1}\} \text{ and } \text{span}\{u\} \oplus \text{span}\{Q_{m-1}\} = \mathcal{H}_m(v_1, A),$$

$\eta_j, j = 1, 2, \ldots, m-1$ are the Ritz values of $A$ from the orthogonal complement $\text{span}\{u\}^\perp$ of $\text{span}\{u\}$ with respect to $\mathcal{H}_m(v_1, A)$. \hspace{1cm} \Box

It is seen from Theorem 3.2 and its proof that the same Krylov subspace $\mathcal{H}_m(v_1, A)$ can be generated by different starting vectors.

As we have seen from Theorems 3.1 and 3.2, the polynomial characterizations of $\bar{\phi}$ and $u$ are related with $v_1$ and $q_1$, respectively. To make them comparable, we naturally hope that both of them could be expressed by either $v_1$ or $q_1$. We will prove that $v_1$ in Theorem 3.1 for $\bar{\phi}$ can be replaced by $q_1$. Conversely, however, it appears that $q_1$ in Theorem 3.2 cannot be replaced by $v_1$.

Note that we have shown $\mathcal{H}_m(q_1, A) = \mathcal{H}_m(v_1, A)$. So any vector $x \in \mathcal{H}_m(v_1, A)$ can be written as $x = t(\lambda)q_1$ with $t(\lambda) \in \mathcal{P}_{m-1}$.

**Theorem 3.3.** Assume that all the Ritz values $\tilde{\lambda}_j, j = 1, 2, \ldots, m$ of $A$ from $\mathcal{H}_m(v_1, A)$ are distinct. Define

$$\bar{\phi} = p(A)q_1.$$ \hspace{1cm} (3.15)

Then

$$p(\lambda) = \beta \prod_{j \neq i} (\lambda - \tilde{\lambda}_j),$$ \hspace{1cm} (3.16)

where $\beta$ is a normalizing factor.
Proof. Since $\mathcal{K}_m(q_1, A) = \mathcal{K}_m(v_1, A)$, $\tilde{\lambda}_j, \tilde{\phi}_j, j = 1, 2, \ldots, m$ are also the Ritz values of $A$ from $\mathcal{K}_m(q_1, A)$. Therefore, $\tilde{\phi} \in \mathcal{K}_m(q_1, A)$ means that there exists a polynomial $p(\lambda)$ with degree $m - 1$ such that

$$\tilde{\phi} = p(A)q_1.$$ 

It thus follows from Theorem 3.1 that the roots of $p(\lambda) = 0$ are the Ritz values $\tilde{\lambda}_j, j \neq i$. $\square$

The following results will help us understand Theorem 3.2.

Lemma 3.1. Let $H_m$ be the Hessenberg matrix in (2.2) and $H_m = U_mT_mU_m^*$ be the Schur decomposition with the entry $t_{11} = \tilde{\lambda}$ and $t_{jj}, j = 2, 3, \ldots, m$ being the other $m - 1$ Ritz values, and $U_m = (x, U_{m-1})$, where $x$ is the eigenvector of $H_m$ associated with $\tilde{\lambda}$. Define $Q_{m-1} = V_mU_{m-1}$. Then the Ritz values of $A$ from $\mathcal{K}_m(v_1, A)$ are the other $m - 1$ Ritz values $\tilde{\lambda}_j, j \neq i$, of $A$ from $\mathcal{K}_m(v_1, A)$.

Proof. We have from the Arnoldi process (2.2) that $V_m^*AV_m = H_m$. By assumption, we may write $H_m = (x, U_{m-1})T_{22}(x, U_{m-1})^*$, where the diagonal entries $T_{22}$ are the $m - 1$ Ritz values $\tilde{\lambda}_j, j \neq i$, and the columns of $U_{m-1}$ are the Schur vectors associated with them.

Now the Ritz values of $A$ from span$\{Q_{m-1}\}$ are the eigenvalues of the matrix $Q_{m-1}^*AQ_{m-1} = U_{m-1}^*H_mU_{m-1} = T_{22}$, which completes the proof. $\square$

Combining Lemma 3.1 with Theorem 3.3, we can derive the following result on $\tilde{\phi}$.

Theorem 3.4. Assume that $\dim(\mathcal{K}_m(v_1, A)) = m$, and let $\mathcal{K}_m(v_1, A) = \text{span}\{\tilde{\phi}\} \oplus \text{span}\{\tilde{\phi}\}^\perp$ be the orthogonal direct sum of $\mathcal{K}_m(v_1, A)$ with $\text{span}\{\tilde{\phi}\}^\perp$ being the orthogonal complement of $\text{span}\{\tilde{\phi}\}$ with respect to $\mathcal{K}_m(v_1, A)$. Define $\tilde{\phi} = p(A)q_1$. (3.17)

Then the roots $\tilde{\lambda}_j, j \neq i$ of $p(\lambda) = 0$ are the Ritz values of $A$ from span$\{\tilde{\phi}\}^\perp \subset \mathcal{K}_m(v_1, A)$.

Proof. From $\tilde{\phi} = V_mx$, it is seen that span$\{\tilde{\phi} \oplus \text{span}\{Q_{m-1}\}\} = \mathcal{K}_m(v_1, A)$. Furthermore, note that $\tilde{\phi} \perp \text{span}\{Q_{m-1}\}$. Therefore, combining with Theorem 3.3, Lemma 3.1 says that the roots of the polynomial characterizing the Ritz
vector \( \tilde{\phi} \) are just the Ritz values of \( A \) from the orthogonal complement \( \text{span}\{\tilde{\phi}\}^\perp = \text{span}\{Q_m\} \) of \( \text{span}\{\tilde{\phi}\} \) with respect to \( \mathcal{H}_m(v_1, A) \).

Recall from the previous notation that
\[
\mathcal{H}_m(v_1, A) = \text{span}\{u\} \oplus \text{span}\{u\}^\perp
= \text{span}\{\tilde{\phi}\} \oplus \text{span}\{\tilde{\phi}\}^\perp.
\]

Therefore, in comparisons with Theorems 3.3 and 3.4, it is seen that Theorem 3.2 is the counterpart of Theorem 3.4 and it gives an analogous result: the roots of the polynomial characterizing the refined Ritz vector \( u \) are the Ritz values of \( A \) from the orthogonal complement \( \text{span}\{u\}^\perp \) of \( \text{span}\{u\} \) with respect to \( \mathcal{H}_m(v_1, A) \). In the sense of the above orthogonal direct sums of \( \mathcal{H}_m(v_1, A) \), Theorems 3.2 and 3.4 have established the correspondence between the polynomial characterizations of \( u \) and \( \tilde{\phi} \).

In terms of the orthogonal direct sum decompositions (3.18) and (3.19), since \( u \) is the best approximation to \( \phi \) from \( \mathcal{H}_m(v_1, A) \) and relation (2.7) holds for \( \|(A - \tilde{\lambda}I)\tilde{\phi}\| \neq 0 \), it is more generally accurate than \( \phi \). This means that the deflated subspace \( \text{span}\{u\}^\perp \) contains richer information on the other eigenvectors \( \phi_j, j \neq i \) than the deflated subspace \( \text{span}\{\tilde{\phi}\}^\perp \) does. It is well known from [8], [27], Ch. 12 and [28], Ch. 4 that the richer information on eigenvectors a subspace contains, the more accurate Ritz values it delivers in general. So \( \eta_j, j = 1, 2, \ldots, m - 1 \) are more accurate approximations to other \( m - 1 \) eigenvalues of \( A \) not equal to \( \lambda \) than \( \tilde{\lambda}_j, j \neq i \), are in general.

We mention that as a by-product we have naturally obtained polynomial characterizations of residuals \( r = (A - \tilde{\lambda}I)u \) computed by the refined Arnoldi method.

**Theorem 3.5.** Define \( r = (A - \tilde{\lambda}I)u \) be the residual of \( \tilde{\lambda}, u \) by the refined Arnoldi method, and define \( r = \tilde{\beta}(A)q_1 \), where \( \tilde{\beta}(\lambda) \in \mathcal{P}_m \) has exact degree \( m \). Then
\[
\tilde{\beta}(\lambda) = \beta(\lambda - \tilde{\lambda}) \prod_{j=1}^{m-1} (\lambda - \eta_j),
\]
where \( \eta_j, j = 1, 2, \ldots, m - 1 \) and \( \beta \) is as before.

4. An implicitly restarted Arnoldi algorithm

We briefly review the implicitly restarted Arnoldi algorithm proposed by Sorensen in [30]. For further information, please see [19].

Assume that \( \lambda_i, \phi_i, i = 1, 2, \ldots, k \) are to be computed. Recall from (2.2) that the \( m \)-step Arnoldi process can be written as
\[
AV_m = V_mH_m + f_m e_m^*.
\]
Assume that \( m - k \) shifts \( \mu_j, j = 1, 2, \ldots, m - k \) are successively applied to \( H_m \), giving
\[
(H_m - \mu_1 I)(H_m - \mu_2 I) \cdots (H_m - \mu_{m-k} I) = QR
\]
with \( Q \) being orthogonal and \( R \) upper triangular.

Let \( H_m^+ = Q^* H_m Q \), \( H_k^+ \) be the \( k \times k \) leading principal submatrix of \( H_m^+ \) and \( V_m^+ = V_m Q = (V_k^+, V_{m-k}^+) \). Then the \( k \)-step Arnoldi process holds
\[
A V_k^+ = V_k^+ H_k^+ + f_k^+ e_k^*,
\]
and it is extended to the \( m \)-step Arnoldi process in a standard way.

The merit of the above scheme is that we need not restart the Arnoldi method from scratch and can save \( k \) matrix-vector products at each restart.

The above scheme involves selection of shifts \( \mu_j, j = 1, 2, \ldots, m - k \). This is the key whether or not an implicitly restarted Arnoldi method is efficient.

It is known from [30] that the updated starting vector \( v_1^+ \) has the form
\[
v_1^+ = \psi(A) v_1
\]
with
\[
\psi(\lambda) = \alpha \prod_{j=1}^{m-k} (\lambda - \mu_j).
\]

Let us expand \( v_1^+ \) as
\[
v_1^+ = \sum_{j=1}^{k} \beta_j \psi(\lambda_j) \varphi_j + \sum_{j=k+1}^{N} \beta_j \psi(\lambda_j) \varphi_j.
\]

Sorensen [30] proved that \( f_k^+ \approx 0 \) if and only if \( v_1^+ \in \text{span}\{\varphi_1, \ldots, \varphi_k\} \). Therefore, numerically, shifts \( \mu_j, j = 1, 2, \ldots, m - k \) are chosen to force \( f_k^+ \approx 0 \) as restart proceeds, so that \( V_k^+ \) spans an approximate invariant subspace of \( A \), i.e., force
\[
v_1^+ \approx \sum_{j=1}^{k} \beta_j \psi(\lambda_j) \varphi_j
\]
and
\[
\sum_{j=k+1}^{N} \beta_j \psi(\lambda_j) \varphi_j \approx 0.
\]

A popular choice for \( \mu_j, j = 1, 2, \ldots, m - k \) is to take them as the unwanted Ritz values \( \tilde{\lambda}_j, j = k+1, \ldots, m \). They are the best approximations available on the unwanted eigenvalues of \( A \) obtained by the Arnoldi method itself, and are called exact shifts [30].

Exact shifts are most widely used, and they often appear to be most efficient for the implicitly restarted Arnoldi algorithm. In [3,4], the Leja shifts were proposed for symmetric matrices. The Leja shifts were often better than the
exact shifts for computing a few smallest eigenvalues when the subspace size is very small. We may simply explain this phenomenon this way: because a subspace of very small size gives only a low degree filter polynomial $\psi(\lambda)$, only a few unwanted eigenvectors could be removed from the updated starting vector $v_1^+$ at each restart. For a subspace of very small size, it may occur that approximations of a few largest eigenvalues of $A$ reappear as eigenvalues of the matrices $H_m$ generated as restart proceeds, that is, unwanted Ritz values vary very little from one restart to next, so that exact shifts are around a few locations in the interval containing the unwanted eigenvalues of $A$ and the other unwanted eigenvectors could not be removed from the updated starting vector $v_1^+$. This may cause the IRA with exact shifts to fail. However, the Leja shifts avoid this phenomenon because selection of new Leja points takes into account the old shifts generated previously, so that the roots of cumulative polynomials are more uniformly distributed within the interval containing the unwanted eigenvalues of $A$ and may thus provide better damping.

5. Implicitly restarting the refined Arnoldi method

We have previously derived the polynomial characterizations of the refined approximate eigenvectors obtained by the refined Arnoldi method. We have analyzed them carefully and proved that the roots of each polynomial are just the Ritz values of $A$ from the orthogonal complement $\text{span}\{u\}^\perp$ of $\text{span}\{u\}$ with respect to $\mathcal{H}_m(v_1, A)$. Furthermore, we have heuristically emphasized the fact that if a refined Ritz vector $u$ is deflated from $\mathcal{H}_m(v_1, A)$ then the Ritz pair of $A$ from the deflated $m - 1$-dimensional subspace are more accurate than the original counterparts obtained from the whole $\mathcal{H}_m(v_1, A)$.

The implicitly restarting scheme of Sorensen is directly applicable to the refined Arnoldi method. The resulting implicitly restarted refined Arnoldi algorithm computes $u_i, i = 1, 2, \ldots, k$ instead of $\phi_i, i = 1, 2, \ldots, k$ as approximate eigenvectors at each restart, and the corresponding residual norms of the approximate eigenpairs are computed by relation (2.6). The key problem we are concerned with is how to select better possible shifts $\mu_j, j = 1, 2, \ldots, m - k$ based on the information available during the refined Arnoldi method.

From the discussions in Section 4, heuristically, the more accurate approximations are the shifts $\mu_j, j = 1, 2, \ldots, m - k$ to some $m - k$ unwanted eigenvalues, the smaller are the components of the corresponding eigenvectors in the term

$$\sum_{j=k+1}^{N} \beta_j \psi(\lambda_j) \phi_j.$$
Numerically, this suggests us to use the best approximations available to some of the unwanted \( \lambda_j, j = k + 1, \ldots, N \) at each restart during an implicitly restarted refined Arnoldi algorithm.

Let us decompose

\[
\mathcal{H}_m(v_1, A) = \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\} \oplus \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\}^\perp. \tag{5.22}
\]

Similar to the proof of Lemma 3.1 and the comments followed, we can give the following result.

**Corollary 5.1.** The exact shifts \( \hat{\lambda}_j, j = k + 1, \ldots, m \) are the Ritz values of \( A \) from the orthogonal complement \( \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\}^\perp \) of \( \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\} \) with respect to \( \mathcal{H}_m(v_1, A) \).

Noting Corollary 5.1, analogously, let us decompose

\[
\mathcal{H}_m(v_1, A) = \text{span}\{u_1, \ldots, u_k\} \oplus \text{span}\{u_1, \ldots, u_k\}^\perp. \tag{5.23}
\]

Then based on the arguments in the end of Section 3, we see that the \( m - k \) Ritz values, say, \( \xi_j, j = 1, 2, \ldots, m-k \), of \( A \) from \( \text{span}\{u_1, \ldots, u_k\}^\perp \) are more accurate than the Ritz values \( \hat{\lambda}_j, j = k + 1, \ldots, m \) of \( A \) from \( \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\}^\perp \) in general because from (2.7) the deflated subspace \( \text{span}\{u_1, \ldots, u_k\}^\perp \) contains more information on the unwanted eigenvectors than the deflated subspace \( \text{span}\{\tilde{\phi}_1, \ldots, \tilde{\phi}_k\}^\perp \). \( \xi_j, j = 1, 2, \ldots, m-k \) are thus the best information available from \( \mathcal{H}_m(v_1, A) \) on the unwanted eigenvalues provided by the refined Arnoldi method itself. So we take \( \xi_j, j = 1, 2, \ldots, m-k \) as shifts for use within an implicitly restarted refined Arnoldi algorithm. \( \xi_j, j = 1, 2, \ldots, m-k \) are called the refined shifts, and they have the same nature as the exact shifts in the sense of (5.22), (5.23) and Corollary 5.1. In other words, for an implicitly restarted refined Arnoldi algorithm, \( \xi_j, j = 1, 2, \ldots, m-k \) have been interpreted to be nothing but the counterparts of the exact shifts for use within an implicitly restarted Arnoldi algorithm.

Next we discuss efficient and reliable computation of \( \xi_j, j = 1, 2, \ldots, m-k \). We give an outline as follows:

- **Step 1:** construct an orthonormal (unitary) basis of \( \text{span}\{u_1, u_2, \ldots, u_k\}^\perp \).
- **Step 2:** project \( A \) onto \( \text{span}\{u_1, u_2, \ldots, u_k\}^\perp \) and form the projected \( (m-k) \times (m-k) \) matrix \( T \).
- **Step 3:** compute the eigenvalues \( \xi_j, j = 1, 2, \ldots, m-k \) of the matrix \( T \).

At a first glance, the computation of \( \xi_i, i = 1, 2, \ldots, m-k \) seems very expensive because Step 1 seems to have to be done at a cost of \( O(N(m-k)^2) \) flops, assuming that a basis of \( \text{span}\{u_1, \ldots, u_k\}^\perp \) is given, and Step 2 seems to need \( 2N\alpha N(m-k) \) flops, assuming that the average non-zero entries of each row of \( A \) is \( N\alpha \). Fortunately, we show that by using some tricks the above process can be realized at a cost of \( O(m^3) \) flops.
We first give details on Step 1. Assume that the sets \{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k\} and 
\{\tilde{\lambda}_{k+1}, \ldots, \tilde{\lambda}_m\} are disjoint. Write \(Z_k = (z_1, z_2, \ldots, z_k)\) with \(z_i, i = 1, 2, \ldots, k\) being those in (2.6). Obviously, \(\text{span}\{V_mZ_k\} = \text{span}\{u_1, \ldots, u_k\}\). We choose a 
\(m \times (m - k)\) matrix \(\hat{Z}_k\) such that the extended \(m \times m\) matrix \(Z = (Z_k, \hat{Z}_k)\) is 
nonsingular. We then compute a QR factorization of \(Z\)

\[
Z = (Z_k, \hat{Z}_k) = (U_k, \hat{U}_k) \begin{pmatrix} R_k & \times \\ \hat{R}_k & \end{pmatrix}
\]

with \(U_k\) and \(\hat{U}_k\) being \(m \times k\) and \(m \times (m - k)\) orthonormal matrices, \(R_k\) and \(\hat{R}_k\) 
being \(k \times k\) and \((m - k) \times (m - k)\) nonsingular upper triangular matrices, respectively and '\times' being a possible non-zero block. Clearly, \(V_mU_k\) is an orthonormal basis of \(\text{span}\{u_1, u_2, \ldots, u_k\}\). We now show that \(V_m\hat{U}_k\) forms 
orthonormal basis of \(\text{span}\{u_1, u_2, \ldots, u_k\}^\perp\): Since \(\text{span}\{V_mZ\} = \text{span}\{V_m\} = \mathcal{N}_m(v_1, A)\), we have

\[
\text{span}\{V_mU_k\} \oplus \text{span}\{V_m\hat{U}_k\} = \text{span}\{u_1, u_2, \ldots, u_k\} \oplus \text{span}\{V_m\hat{U}_k\} = \mathcal{N}_m(v_1, A).
\]

Now because

\[
\text{span}\{V_m\hat{U}_k\} \perp \text{span}\{V_mU_k\} (= \text{span}\{u_1, u_2, \ldots, u_k\}),
\]

we have

\[
\text{span}\{V_m\hat{U}_k\} = \text{span}\{u_1, u_2, \ldots, u_k\}^\perp,
\]

which shows that \(V_m\hat{U}_k\) is an orthonormal basis of \(\text{span}\{u_1, u_2, \ldots, u_k\}^\perp\).

Two notes on implementation come in order: first, if the matrix \(A\) is real and 
some \(\tilde{\lambda}_i, i = 1, 2, \ldots, k\) is complex, then we assume the complex conjugate of \(\tilde{\lambda}_i\) 
to be in the set \(\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k\}\) and take the real and imaginary parts of the corresponding \(z_i\) as two columns of the above matrix \(Z_k\); second, the above 
\(m \times (m - k)\) matrix \(\hat{Z}_k\) is generated randomly in a uniform distribution. Such a 
way ensures that \(Z\) is nonsingular in practice. The total cost of computing \(\hat{U}_k\) is 
\(O(m^3)\).

We now discuss Steps 2 and 3. We have from (2.2) that \(H_m = V_m^*AV_m\). 
Therefore, the projected matrix \(T\) at Step 2 can be efficiently computed by

\[
T = (V_m\hat{U}_k)^*A(V_m\hat{U}_k) = \hat{U}_k^*H_m\hat{U}_k.
\]

So it is not necessary to form \(V_m\hat{U}_k\) explicitly at Steps 1 and 2, and Step 2 costs \(2km^2\) flops. Step 3 is direct, and it costs about \(9(m - k)^3\) flops if the QR 
algorithm is used. As a whole, the total cost of computing the shifts \(\xi_j, j = 1, 2, \ldots, m - k\) is no more than \(13m^3\) flops usually, assuming that the 
Gram–Schmidt orthogonalization with one step iterative refinement is used at 
Step 1. If Householder transformations are used instead, then Step 1 costs 
\(4m^3/3\) flops. So it is cheaper than the Gram–Schmidt orthogonalization.
Finally, we have three comments: first, in exact arithmetic, $\xi_j$, $j = 1, 2, \ldots, m - k$ are independent of $U_k$ provided that the above matrix $Z_k$ ensures that $Z$ is nonsingular; second, since $T$ is real, the $QR$ iteration (4.21) can be successively performed only in real arithmetic even though $T$ has complex eigenvalues; third, noting that the refined Arnoldi method uses $k$ small sized SVDs to compute $z_1, z_2, \ldots, z_k$ and the computational cost is $O(km^3)$ flops while the Arnoldi method only costs $O(m^3)$ flops to compute the eigenpairs $\hat{\lambda}_j, x_j, j = 1, 2, \ldots, m$ of $H_m$, so the refined Arnoldi method is a little more expensive than the Arnoldi method at each restart.

6. Numerical experiments

We report numerical experiments on four problems. We have tested the algorithm of Sorensen and ours using MATLAB on an Intel Pentium 100 MHZ with extended memory 40 Megabytes. We use the function eigs.m, the MATLAB counterpart of ARPACK. Our algorithm is different from the eigs.m in that the exact shifts were replaced by the refined shifts and the Ritz vectors used to approximate the $k$ wanted eigenvectors were replaced by the corresponding refined Ritz vectors. Note that in order to speed up convergence, $k + 3$ Ritz pairs are computed to approximate $k + 3$ eigenpairs, and $m - (k + 3)$ shifts are used in the eigs.m. Our algorithm has adopted this strategy.

Example 1. This problem is from [2], and it models the concentration waves for reaction and transport interaction of chemical solutions in a tubular reactor. The concentrations $x(t, z)$ and $y(t, z)$ of two reacting diffusing components are modeled by the system

\[
\begin{align*}
\frac{\partial x}{\partial t} & = \frac{\delta_1}{L^2} \frac{\partial^2 x}{\partial z^2} + f(x, y), \\
\frac{\partial y}{\partial t} & = \frac{\delta_2}{L^2} \frac{\partial^2 y}{\partial z^2} + g(x, y)
\end{align*}
\]  

(6.24)

(6.25)

with the initial conditions $x(0, z) = x_0(z)$, $y(0, z) = y_0(z)$ and the Dirichlet boundary conditions $x(t, 0) = x(t, 1) = x^*$, $y(t, 0) = y(t, 1) = y^*$, where $0 \leq z \leq 1$ is the space coordinate along the tube, and $t$ is time. In particular, one is interested in the Brusselator wave model in which

\[ f(x, y) = \alpha - (\beta + 1)x + x^2y, \quad g(x, y) = \beta x - x^2y. \]

Then the above system admits the stationary solution $x^* = \alpha$, $y^* = \beta/\alpha$. In this problem, one is primarily interested in the existence of stable periodic solutions to the system as the bifurcation parameter $L$ varies. This occurs when
the eigenvalues with largest real parts of the Jacobian of the right-hand sides of (6.24) and (6.25), evaluated at the steady station solution, is purely imaginary. To verify this fact numerically, one first needs to discretize the equations with respect to \( z, t \) and compute the eigenvalues with largest real parts of the resulting discrete Jacobian.

If we discretize the interval \([0,1]\) using \( n \) interior points with the uniform mesh size \( h = 1/(n+1) \), then the discretized Jacobian of the system is a \( 2 \times 2 \) block matrix of the form

\[
A = \begin{pmatrix}
\tau_1 T + (\beta - 1)I & \alpha^2 I \\
-\beta I & \tau_2 T - \alpha^2 I
\end{pmatrix},
\]

where \( A \) is of order \( N = 2n \) and \( T = \text{tridiag}\{1,-2,1\}, \tau_1 = \delta_1/h^2 L^2 \) and \( \tau_2 = \delta_2/h^2 L^2 \).

We tested the matrix \( A \) of order \( N = 2000 \) using the two algorithms, and the data file is BW2000 from [2], where \( \delta_1 = 0.008, \delta_2 = \frac{1}{2} \delta_1 = 0.004, \alpha = 2, \beta = 5.45, L \approx 0.51302 \). We computed the five rightmost eigenpairs, and the algorithms stopped as soon as maxima of (relative) residual norms dropped below \( tol = 1.0 \times 10^{-6} \). We used the same starting vector generated randomly in a uniform distribution in the two algorithms. In all tables of the paper, IRA and IRRA denote the implicitly restarted Arnoldi method and implicitly restarted refined Arnoldi method, respectively, ‘iter’ denotes the number of restarts, ‘CPU’ is the CPU timings in seconds, ‘mv’ is the matrix-vector products accessing the matrix \( A \) and ‘Res. norms’ denotes max\(1 \leq i \leq k\)\(\|\|A - \lambda_i I\||\phi_i\|/\|A\|\|\) or max\(1 \leq i \leq k\)\(\|\|A - \lambda_i I\||u_i\|/\|A\|\|\). Table 1 reports the numerical results, where the symbol ‘n.c’ denotes no convergence. The computed five eigenvalues are \( \lambda_{1,2} \approx 2.4427 \times 10^{-7} \pm 2.13950913i, \lambda_{3,4} \approx -0.674996807 \pm 2.52870849i \) and \( \lambda_5 \approx -1.79998450 - 3.03273199i \).

Table 1 shows that the IRRA performed better than the IRA in terms of CPU timings, restarts and matrix–vector products. The IRRA was about one and a half times as fast as the IRA was. The results indicate that we benefit by

---

**Table 1**

<table>
<thead>
<tr>
<th>Example 1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>( m )</th>
<th>IRA</th>
<th></th>
<th>IRRA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>CPU</td>
<td>mv</td>
</tr>
<tr>
<td>10</td>
<td>&gt;2000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>1428</td>
<td>7628</td>
<td>19149</td>
</tr>
<tr>
<td>30</td>
<td>449</td>
<td>5931</td>
<td>10486</td>
</tr>
<tr>
<td>40</td>
<td>218</td>
<td>5445</td>
<td>7249</td>
</tr>
<tr>
<td>50</td>
<td>138</td>
<td>5487</td>
<td>5999</td>
</tr>
</tbody>
</table>
using the refined Ritz vectors and the refined shifts implicitly applied. We also see that for \( m = 10 \) both the IRA and the IRRA did not converge. This is, of course, not surprising because there was no reason to expect that updating subspaces of very small size would contain enough information on the required eigenvectors as restart proceeded. In experiments, we labeled both the exact shifts, rewritten as \( \gamma_j = \lambda_{k+3+j} \), and the refined shifts \( \xi_j, j = 1, m - k - 3 \) in the same order at each restart of the IRRA. Define the relative difference \( e_{r,j} = |\gamma_j - \xi_j|/|\gamma_j| \) of \( \gamma_j \) and \( \xi_j \). Fig. 1 plotted the curve of \( \max_{1 \leq j \leq m-k-3} e_{r,j} \) as restarts proceeded for Example 1 and \( m = 30 \) using the IRRA. We observe from the figure that the refined shifts were essentially different from the exact shifts in the same subspace at many restarts. This means that the refined Ritz vectors and the refined shifts improved the corresponding Ritz vectors and the exact shifts significantly in the same subspace at many restarts, so that the IRRA was considerably faster than the IRA.

**Example 2.** Dielectric channel waveguide problems arise in many integrated circuit applications. Discretization of the governing Helmholtz equation for the magnetic field \( H \),

\[
\nabla^2 H_x + k^2 n(x,y) H_x = \beta^2 H_x,
\]

\[
\nabla^2 H_y + k^2 n(x,y) H_y = \beta^2 H_y
\]

![Fig. 1. Example 1 using the IRRA for m = 30.](image-url)
by finite difference leads to a nonsymmetric matrix eigenvalue problem of the form
\[
\begin{pmatrix}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{pmatrix}
\begin{pmatrix}
H_x \\
H_y
\end{pmatrix}
= \beta^2
\begin{pmatrix}
B_{11} \\
B_{22}
\end{pmatrix}
\begin{pmatrix}
H_x \\
H_y
\end{pmatrix},
\]
where $C_{11}$ and $C_{22}$ are five- or tri-diagonal matrices, $C_{12}$ and $C_{21}$ are (tri-)diagonal matrices, and $B_{11}$ and $B_{22}$ are nonsingular diagonal matrices.

The computational task is to compute the rightmost eigenpairs. There are eigenvalues with negative real part several orders of magnitude larger than the desired eigenvalues with positive real part, and also the desired eigenvalues are clustered for large $N$. As was pointed out in [2], this problem presents a challenge to existing numerical methods.

We tested the problem with $N = 2048, 8192$, and the data files were got by ftp from [2]. We computed the five rightmost eigenpairs. The same randomly generated starting vector was used, and the algorithms stopped once residual norms dropped below to1 = $1 \times 10^{-6}$. Tables 2 and 3 report the results obtained. The computed five eigenvalues for $N = 2048, 8192$ are $\lambda_1 \approx 0.9788023$, $\lambda_2 \approx 0.97780460$, $\lambda_3 \approx 0.96551437$, $\lambda_4 \approx 0.96473889$, $\lambda_5 \approx 0.94725160$ and $\lambda_1 \approx 0.86657564$, $\lambda_2 \approx 0.86596590$, $\lambda_3 \approx 0.86590932$, $\lambda_4 \approx 0.86568081$, $\lambda_5 \approx 0.86549610$, respectively.

For this problem, the IRRA was more efficient than the IRA. Obviously, the problem of order $N = 8192$ is much harder to solve than that of order $N = 2048$. It is seen from Table 3 that both the IRA and the IRRA used many restarts to achieve the desired accuracy. But the IRRA was faster than the IRA. Fig. 2 plotted the curve of the maximum relative difference $\max_{1 \leq j \leq m-k-3} |e_{r,j}|$ of the exact shifts with the refined shifts at each restart of the IRRA for $N = 2048, m = 20$. We observe from the figure that the refined shifts were different from the exact shifts at many restarts.

Example 3. This matrix comes from well-known random walk problems; for details, see [2]. We took the order $N = 5151$ and computed the three rightmost

<table>
<thead>
<tr>
<th>$m$</th>
<th>IRA</th>
<th>IRRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter</td>
<td>CPU</td>
<td>mv</td>
</tr>
<tr>
<td>10</td>
<td>2000</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>47</td>
<td>208</td>
</tr>
<tr>
<td>30</td>
<td>18</td>
<td>233</td>
</tr>
<tr>
<td>40</td>
<td>11</td>
<td>270</td>
</tr>
<tr>
<td>50</td>
<td>8</td>
<td>315</td>
</tr>
<tr>
<td>60</td>
<td>7</td>
<td>408</td>
</tr>
</tbody>
</table>
Table 3
Example 2, N = 8192

<table>
<thead>
<tr>
<th>m</th>
<th>IRA</th>
<th>IRRA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>CPU</td>
</tr>
<tr>
<td>20</td>
<td>&gt;2000</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>530</td>
<td>27233</td>
</tr>
<tr>
<td>40</td>
<td>206</td>
<td>18547</td>
</tr>
</tbody>
</table>

eigenpairs. We took \( tol = 1.\times 10^{-8} \) and the same starting vector was used.

Table 4 gives the numerical results.

From Table 4, it is seen that both the IRA and IRRA are quite efficient for this problem, but the latter was faster. Fig. 3 plotted the curve of the maximum of relative errors of the exact shifts and the refined shifts at each restart for \( m = 10 \) using the IRRA. We observe from the figure that the refined shifts were quite different from the exact shifts in the same subspace at many restarts. This means that the refined Ritz vectors improved the corresponding Ritz vectors significantly and the refined shifts are much better than the exact shifts, so that the IRRA behaved different from the IRA and the former was much faster than the latter.

![Fig. 2. Example 2 using the IRRA for \( m = 20 \).](image-url)
Table 4
Example 3

<table>
<thead>
<tr>
<th>m</th>
<th>IRA iter</th>
<th>IRA CPU</th>
<th>IRA mv</th>
<th>IRA Res. norms</th>
<th>IRRA iter</th>
<th>IRRA CPU</th>
<th>IRRA mv</th>
<th>IRRA Res. norms</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>204</td>
<td>398</td>
<td>695</td>
<td>$9.7 \times 10^{-9}$</td>
<td>94</td>
<td>243</td>
<td>393</td>
<td>$5.5 \times 10^{-9}$</td>
</tr>
<tr>
<td>20</td>
<td>31</td>
<td>439</td>
<td>420</td>
<td>$4.1 \times 10^{-9}$</td>
<td>26</td>
<td>381</td>
<td>385</td>
<td>$9.5 \times 10^{-9}$</td>
</tr>
<tr>
<td>30</td>
<td>23</td>
<td>801</td>
<td>564</td>
<td>$6.1 \times 10^{-9}$</td>
<td>20</td>
<td>728</td>
<td>489</td>
<td>$6.1 \times 10^{-9}$</td>
</tr>
<tr>
<td>40</td>
<td>18</td>
<td>1178</td>
<td>602</td>
<td>$8.8 \times 10^{-9}$</td>
<td>15</td>
<td>1025</td>
<td>519</td>
<td>$4.1 \times 10^{-9}$</td>
</tr>
<tr>
<td>50</td>
<td>18</td>
<td>2049</td>
<td>824</td>
<td>$5.7 \times 10^{-9}$</td>
<td>13</td>
<td>1415</td>
<td>581</td>
<td>$3.8 \times 10^{-9}$</td>
</tr>
</tbody>
</table>

**Example 4.** This test matrix is from the following constant-coefficient convection–diffusion equation

$$- \Delta u(x, y) + p_1 u_x(x, y) + p_2 u_y(x, y) - p_3 u(x, y) = f(x, y)$$

defined on the unit square region $[0, 1] \times [0, 1]$ with the boundary condition $u(x, y) = 0$ and $p_1, p_2$ and $p_3$ being positive constants. Discretization by five point difference on a uniform $n \times n$ grid and numbering the grid points using the rowwise natural ordering give a block tridiagonal matrix of the form

![Fig. 3. Example 3 using the IRRA for $m = 10$.](image-url)
\[
A = \begin{pmatrix}
T & (\beta + 1)I \\
(\beta + 1)I & T
\end{pmatrix}
\]

with
\[
T = \begin{pmatrix}
4 - \sigma & \gamma - 1 \\
-\gamma - 1 & 4 - \sigma & \gamma - 1 \\
& & \ddots & \ddots & \ddots \\
& & & \gamma - 1 \\
& & & -\gamma - 1 & 4 - \sigma
\end{pmatrix}
\]

where \( \beta = \frac{p_1 h}{2} \), \( \gamma = \frac{p_2 h}{2} \), \( \sigma = \frac{p_3 h}{2} \) and \( h = \frac{1}{n + 1} \). The order of \( A \) is \( N = n^2 \).

We tested \( A \) with \( p_1 = 1, p_2 = p_3 = 0 \) and \( N = 900 \) using the IRA and the IRRA. We computed the three rightmost eigenpairs and the stopping criterion was \( tol = 1.E - 10 \). The resulting matrix \( A \) is nearly symmetric and was ever tested in Ref. \[9\]. Table 5 displays the results obtained, and the computed eigenvalues are \( \lambda_1 \approx 7.97921847, \lambda_2 \approx 7.94854369 \) and \( \lambda_3 \approx 7.94839701 \). In the table, the results in braces were the output for \( tol = 1.25 \times 10^{-9} \), they will be used for a comparison with the algorithm in \[9\] later.

It is seen from Table 5 that both the IRA and the IRRA solved the problem very efficiently. Also, they are much more efficient than the explicitly restarting scheme used in \[9\] adapted from Saad \[28\], p. 234. In \[9\], the absolute residual norms of the approximate eigenpairs were required to be less than or equal to \( 10^{-8} \). This is equivalent to using the stopping criterion \( tol = 10^{-8}/\|A\|_1 = 1.25 \times 10^{-9} \) in the IRA and the IRRA. For such a \( tol \) and \( m \geq 60 \), the IRA and the IRRA used the same the number of restarts and matrix-vector products as those for \( tol = 1.E - 10 \) when taking \( m = 60, 80, 90 \), but they produced different output for \( m = 70, 100 \), as shown in the braces of Table 5. We can see from \[9\] that explicitly restarting the refined Arnoldi method with Saad's scheme used 3240, 1610, 1520, 450 and 300 matrix-vector products, respectively, for \( m = 60, 70, 80, 90, 100 \). In comparisons with Table 5, we observe that both the IRA and the IRRA could be up to nearly 14 times (i.e., for \( m = 60 \)) faster than the algorithm in \[9\]. For this matrix, although for most \( m \) the IRRA used no more restarts than the IRA, the IRRA had no advantage over the IRA because the IRRA used a little bit more CPU time than the IRA at each restart. Occasionally, the number of restarts used by the IRRA was more than that used by the IRA, e.g., \( m = 40 \). This is not surprising in
Table 5
Example 4

<table>
<thead>
<tr>
<th>m</th>
<th>IRA</th>
<th>IRRA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>CPU</td>
</tr>
<tr>
<td>10</td>
<td>102</td>
<td>70.8</td>
</tr>
<tr>
<td>20</td>
<td>17</td>
<td>55.3</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>75.0</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
<td>78.5</td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>106</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
<td>117</td>
</tr>
<tr>
<td>70</td>
<td>4</td>
<td>163</td>
</tr>
<tr>
<td>(70)</td>
<td>(3)</td>
<td>(115)</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
<td>152</td>
</tr>
<tr>
<td>90</td>
<td>3</td>
<td>195</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>248</td>
</tr>
<tr>
<td>(100)</td>
<td>(2)</td>
<td>(139)</td>
</tr>
</tbody>
</table>

Numerical computation because the refined Arnoldi method is better than the Arnoldi method in the same subspace but this global optimality is lost in the restarted sense and only local optimality exists. Fig. 4 plotted the curve of the maximum of relative differences between the exact shifts and the refined shifts.

![Fig. 4. Example 4 using the IRRA for m = 10.](image)
at each restart for \( m = 10 \) using the IRRA. We observe from the figure that the refined shifts were very close to the exact shifts in the same subspace at most of restarts, that is, the refined Ritz vectors were very close to the corresponding Ritz vectors in the same subspace at most of restarts, so that the IRRA behaved almost the same as the IRA.

We also tested the matrix \( A \) with \( p_1 = 25, p_2 = 50 \) and \( p_3 = 300 \) and computed the three rightmost eigenpairs with the stopping criterion \( \text{tol} = 1.E - 10 \). The resulting matrix \( A \) is point difference highly non-normal. Unlike all the previous examples, we found that both the IRA and the IRRA were quite sensitive to a starting vector. In any event, both the IRA and the IRRA used much more restarts to converge for this matrix than those for the matrix with \( p_1 = 1, p_2 = p_3 = 0 \). We ran the IRA and the IRRA on such a matrix using ten starting vectors generated randomly in a uniform distribution, and we found that in most cases (for eight of the chosen ten starting vectors) the IRRA was considerably faster than the IRA for the same starting vector and the former was often about twice as fast as the latter in terms of restarts, CPU timings and matrix-vector products. For each of the other two starting vectors, the IRRA was a little slower than the IRA for some \( m \).

Acknowledgements

I thank the referees for their valuable suggestions and detailed comments, which led me to simplify the presentation. In particular, I would like to express my deep thanks to Rich Lehoucq for many stimulating discussions and constructive suggestions that improved on the paper significantly.

References
