East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.070517.180917a

Vol. **8**, No. 1, pp. 82-99
February 2018

Implicitly Restarted Refined Generalised Arnoldi Method with Deflation for the Polynomial Eigenvalue Problem

Wei Wei and Hua Dai*

Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China.

Received 7 May 2017; Accepted (in revised version) 18 September 2017.

Abstract. Based on the generalised Arnoldi procedure, we develop an implicitly restarted generalised Arnoldi method for solving the large-scale polynomial eigenvalue problem. By combining implicit restarting with the refinement scheme, we present an implicitly restarted refined generalised Arnoldi (IRGAR) method. To avoid repeated converged eigenpairs in the later iteration, we develop a novel non-equivalence low-rank deflation technique and propose a deflated and implicitly restarted refined generalised Arnoldi method (DIRGAR). Some numerical experiments show that this DIRGAR method is efficient and robust.

AMS subject classifications: 65F15

Key words: Polynomial eigenvalue problem, generalised Arnoldi method, refinement, implicit restarting, non-equivalence low-rank deflation.

1. Introduction

We consider the matrix polynomial

$$P(\lambda) = \lambda^d A_d + \lambda^{d-1} A_{d-1} + \cdots + \lambda A_1 + A_0, \tag{1.1}$$

where the coefficient matrices $A_i (0 \le i \le d)$ are $n \times n$ large and sparse. The scalar λ is said to be an eigenvalue of $P(\lambda)$ if $det(P(\lambda)) = 0$, where $det(P(\lambda))$ denotes the determinant of the matrix $P(\lambda)$. The nonzero vectors x, y are said to be the right and left eigenvectors corresponding to the eigenvalue λ if

$$P(\lambda)x = 0 \quad \text{and} \quad y^H P(\lambda) = 0\,, \tag{1.2}$$

respectively. This is the well known polynomial eigenvalue problem (PEP). For convenience, the two-tuple (λ, x) or the triplet (λ, x, y) is used to denote an eigenpair of the problem (1.2).

*Corresponding author. *Email addresses:* w.wei@nuaa.edu.cn (W. Wei), hdai@nuaa.edu.cn (H. Dai)

The PEP reduces to the generalised eigenvalue problem if $d = 1$; and to the quadratic eigenvalue problem (QEP) if $d = 2$, which is one of the most important cases. Thus for the QEP Tisseur & Meerbergen [31] surveyed many applications, properties and numerical methods. The cubic eigenvalue problem (CEP) when $d = 3$ arises in the numerical simulation of the semiconductor quantum dot model [15, 32]; and higher order polynomial eigenvalue problems arise in stability analysis in control systems [12], the simulation of the three-dimensional pyramid quantum dot heterostructure [16], and in structural dynamic analysis via the dynamic element method [26] or the least squares element method [27].

Gohberg *et al.* [9] established the mathematical theory for matrix polynomials. Chu [6], Dedieu & Tisseur [7], Higham & Tisseur [13] considered perturbation analysis for the PEP. Tisseur [30], and Higham *et al.* [10] and Lawrence & Corless [21] analysed the backward error. Here we consider the computation of some eigenpairs with eigenvalues of largest modulus. If some eigenpairs with eigenvalues nearest to a target σ are desired, one may apply the shift-invert transformation $\tilde{\lambda} = 1/(\lambda - \sigma)$ to the matrix polynomial $P(\lambda)$ and consider the transferred matrix polynomial

$$\tilde{P}(\tilde{\lambda}) = \tilde{\lambda}^d \tilde{A}_d + \tilde{\lambda}^{d-1}\tilde{A}_{d-1} + \cdots + \tilde{\lambda}\tilde{A}_1 + \tilde{A}_0 \,, \tag{1.3}$$

where $\tilde{A}_i = \sum_{j=0}^{i} C_{d-j}^{i-j} \sigma^{i-j} A_{d-j} (0 \le i \le d)$, $C_n^k = \frac{n!}{k!(n-k)!}$. For $\sigma = 0$ (some smallest modulus eigenvalues are desired), such that the new coefficient matrices satisfy $\tilde{A}_i = A_{d-i}$ $(0 \le i \le d)$, one simply inverts the order of the coefficient matrices $A_i (0 \le i \le d)$ in $P(\lambda)$. If the coefficient matrix A_d is singular, then $\lambda = \infty$ is an eigenvalue of the PEP (1.2) — and here we assume that A_d is nonsingular throughout.

The classical approach for solving the PEP (1.2) is to linearise the problem and produce the following generalised eigenvalue problem:

$$Cy = \lambda G y \,, \tag{1.4}$$

where

$$C = \begin{pmatrix} -A_{d-1} & -A_{d-2} & \cdots & -A_0 \\ I & & & \\ & \ddots & & \\ & & I & 0 \end{pmatrix}, \qquad G = \begin{pmatrix} A_d & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix},$$

$$y = \begin{pmatrix} \lambda^{d-1} x \\ \lambda^{d-2} x \\ \vdots \\ x \end{pmatrix}. \tag{1.5}$$

Many different linearisations are possible [1, 9, 11, 23, 24]. Provided A_d is nonsingular, the generalised eigenvalue problem (1.4) may be reduced to the standard eigenvalue problem

$$My = \lambda y, \tag{1.6}$$

where

$$M = G^{-1}C = \begin{pmatrix} M_{d-1} & M_{d-2} & \cdots & M_0 \\ I & & & \\ & \ddots & & \\ & & I & 0 \end{pmatrix},$$

$$M_i = -A_d^{-1}A_i \ (0 \le i \le d-1)\,. \tag{1.7}$$

The linearisation process increases the problem size, and the matrix structures and certain properties of the original PEP are not preserved. Thus additional memory is required for any resulting numerical solution, but more importantly the linearised eigenvalue problems (1.4) and (1.6) become more ill-conditioned. To avoid these disadvantages, many researchers have been studying numerical methods to solve the PEP (1.2) directly. Sleijpen *et al.* [28] discussed the Jacobi-Davidson method for the PEP (1.2), and Hwang *et al.* [17] developed a parallel Jacobi-Davidson approach for the large sparse polynomial eigenvalue problem. The Jacobi-Davidson method targets one eigenpair at a time with local convergence, but if the desired eigenvalues of the PEP (1.2) are clustered near each other it has difficulties in detecting and resolving them. Meerbergen [25] proposed a quadratic Arnoldi (Q-Arnoldi) method for the solution of the QEP, but it may suffer from numerical instabilities. Recently, Lu *et al.* [22] presented a two-level orthogonal Arnoldi (TOAR) procedure that avoids the instabilities of the Q-Arnoldi method, and Kressner & Roman [20] extended the Q-Arnoldi and TOAR algorithms to the PEP in an elegant way. Based on the Hessenberg-triangular decomposition of a matrix pencil, Huang *et al.* [14] proposed a semiorthogonal generalised Arnoldi method for solving the QEP, and this method was recently extended for the PEP by Wei & Dai [33]. Bai & Su [3] proposed a second-order Arnoldi (SOAR) method for the QEP, and Bao *et al.* [4] later extended this method to the generalised Arnoldi (GAR) for solving the higher order polynomial eigenvalue problem. Bao *et al.* [4] developed an explicitly restarted refined generalised Arnoldi (RGAR) method for the PEP based on the refined projection principle [18], but implicit restarting was not considered.

In Section 2, we give a brief description of the explicitly restarted generalised Arnoldi (GAR) method and its refinement scheme (RGAR). The implicit restarting technique [19] for the SOAR method led us to develop the implicitly restarted version of the GAR method discussed in Section 3, on combining the implicit restarting strategy with the refinement scheme for the PEP — i.e. our implicitly restarted refined generalised Arnoldi (IRGAR) method. The implicit deflation technique based on Schur forms (e.g. see Ref. [2]) combined with some eigen-solver procedure performs well for the linear eigenvalue problem, but for the PEP a Schur form does not exist. Similar to the non-equivalence low-rank deflation technique presented by Wei & Dai [33], for the PEP we develop a non-equivalence low-rank deflation technique to provide a deflated and implicitly restarted refined generalised Arnoldi method (DIRGAR). Our implicitly restarted refined generalised Arnoldi method with the non-equivalence low-rank deflation is developed in Section 4 for the polynomial eigenvalue problem. Numerical examples are reported in Section 5, and some concluding remarks are made in Section 6.

We use the following notation throughout. The $n \times n$ identity matrix is denoted by I_n, or I for short, and e_j denotes the j-th column of the identity matrix. Superscripta T and H respectively denote the transpose and conjugate transpose of a vector or a matrix, $\|\cdot\|_2$ the Euclidean vector norm or the induced matrix norm, $\|\cdot\|_F$ the Frobenius matrix norm, $diag_r(\lambda_1, \lambda_2, \cdots, \lambda_r)$ a diagonal matrix with the entries $\lambda_i (1 \le i \le r)$, and the subindex r in $diag_r$ denotes the dimension of the diagonal matrix.

2. The Generalised Arnoldi Method

As background, we briefly describe the generalised Arnoldi procedure, the explicitly restarted generalised Arnoldi (GAR) method, and the refinement scheme (RGAR) for the PEP [4].

Definition 2.1 (cf. Ref. [4]). Let $M_0, M_1, \cdots, M_{d-1}$ be the given matrices of order n and $u_0, u_1, \cdots, u_{d-1}$ be n-dimensional vectors with $u_{d-1} \neq 0$, such that

$$\begin{aligned} r_0 &= u_1 , \\ r_1 &= u_2 , \\ &\vdots \\ r_{d-2} &= u_{d-1} , \\ r_{d-1} &= M_{d-1}u_{d-1} + M_{d-2}u_{d-2} + \cdots + M_0 u_0 , \\ r_j &= M_{d-1}r_{j-1} + M_{d-2}r_{j-2} + \cdots + M_0 r_{j-d} , \ \ j \ge d . \end{aligned}$$

Then the sequence $\{r_i\}_{i=0}^{j}$ is called a generalised Krylov sequence based on $\{M_i\}_{i=0}^{d-1}$ and $\{u_i\}_{i=0}^{d-1}$, and the space

$$K_m\left[\{M_i\}_{i=0}^{d-1}, \{u_i\}_{i=0}^{d-1}\right] = span\{r_{d-2}, r_{d-1}, \cdots, r_{d+m-3}\} \tag{2.1}$$

is called an m-th generalised Krylov subspace.

Let $v = (u_{d-1}^H, u_{d-2}^H, \cdots, u_1^H, u_0^H)^H \in \mathscr{C}^{dn}$, when

$$K(M, v, m) = span\{v, Mv, \cdots, M^{m-1}v\} \tag{2.2}$$

is the Krylov subspace for the standard eigenvalue problem (1.6) and the following relation holds:

$$\begin{pmatrix} r_j \\ r_{j-1} \\ \vdots \\ r_{j-d+1} \end{pmatrix} = M^{j-d+2}v , \ \ j \ge d-1 . \tag{2.3}$$

The relation (2.3) shows that one may directly solve the PEP (1.2) using the Krylov subspace (2.1), instead of using the subspace (2.2) for the standard eigenvalue problem (1.6).

We have the following algorithm for generating an orthonormal basis of the subspace (2.1).

Algorithm 2.1. Generalised Arnoldi Method (cf. Ref. [4]).

Input: matrices $\{M_i\}_{i=0}^{d-1}$, initial vectors $\{u_i\}_{i=0}^{d-1}$ with $u_{d-1} \neq 0$ and dimension m of subspace.
Output: vectors $\{q_i\}_{i=1}^{m+1}$, $\{p_1^{(i)}, p_2^{(i)}, \cdots, p_m^{(i)}, p_{m+1}^{(i)}\}_{i=1}^{d-1}$, and scalars $\{h_{i,j}\}$.

1. $\beta = \|u_{d-1}\|_2, \begin{pmatrix} q_1 \\ p_1^{(1)} \\ \vdots \\ p_1^{(d-1)} \end{pmatrix} = \frac{1}{\beta} \begin{pmatrix} u_{d-1} \\ u_{d-2} \\ \vdots \\ u_0 \end{pmatrix}$;

2. **for** $j = 1 : m$ **do**
$$\begin{pmatrix} q_{j+1} \\ p_{j+1}^{(1)} \\ p_{j+1}^{(2)} \\ \vdots \\ p_{j+1}^{(d-1)} \end{pmatrix} = \begin{pmatrix} M_{d-1}q_j + M_{d-2}p_j^{(1)} + \cdots + M_0 p_j^{(d-1)} \\ q_j \\ p_j^{(1)} \\ \vdots \\ p_j^{(d-2)} \end{pmatrix},$$
for $i = 1 : j$ **do**
$h_{i,j} = q_i^H q_{j+1}$,
$$\begin{pmatrix} q_{j+1} \\ p_{j+1}^{(1)} \\ \vdots \\ p_{j+1}^{(d-1)} \end{pmatrix} = \begin{pmatrix} q_{j+1} \\ p_{j+1}^{(1)} \\ \vdots \\ p_{j+1}^{(d-1)} \end{pmatrix} - h_{i,j} \begin{pmatrix} q_i \\ p_i^{(1)} \\ \vdots \\ p_{ji}^{(d-1)} \end{pmatrix},$$
endfor
$h_{j+1,j} = \|q_{j+1}\|_2$,
if $h_{j+1,j} = 0$, **then** stop;
else
$$\begin{pmatrix} q_{j+1} \\ p_{j+1}^{(1)} \\ \vdots \\ p_{j+1}^{(d-1)} \end{pmatrix} = \frac{1}{h_{j+1,j}} \begin{pmatrix} q_{j+1} \\ p_{j+1}^{(1)} \\ \vdots \\ p_{j+1}^{(d-1)} \end{pmatrix}.$$
endfor

If Algorithm 2.1 stops prematurely at step $j < m$, then either *breakdown* or *deflation* occurs at the step. If breakdown occurs, the algorithm just needs to stop, but if deflation occurs some remedies should be considered — cf. Refs. [3, 19] for details. However, here we simply assume that Algorithm 2.1 just stops at step m.

Let $Q_m = [q_1, q_2, \cdots, q_m]$ and $P_m^{(i)} = [p_1^{(i)}, p_2^{(i)}, \cdots, p_m^{(i)}]$; and let $\tilde{H}_m = (h_{i,j})$ denote the $(m+1)\times m$ upper Hessenberg matrix whose nonzero entries are generated by Algorithm 2.1, and let H_m be the $m \times m$ matrix obtained from $\tilde{H}_m$ by deleting the last row. Bao *et al.* [4]

proved that

$$M\begin{pmatrix} Q_m \\ P_m^{(1)} \\ \vdots \\ P_m^{(d-1)} \end{pmatrix} = \begin{pmatrix} Q_m \\ P_m^{(1)} \\ \vdots \\ P_m^{(d-1)} \end{pmatrix} H_m + h_{m+1,m}\begin{pmatrix} q_{m+1} \\ p_{m+1}^{(1)} \\ \vdots \\ p_{m+1}^{(d-1)} \end{pmatrix} e_m^T , \tag{2.4}$$

and that $\{q_i\}_{i=1}^m$ is an orthonormal basis of the generalised Krylov subspace (2.1).

Via the Rayleigh-Ritz projection technique on the subspace $\mathscr{Q}_m = span\{Q_m\}$, Bao *et al.* [4] developed a generalised Arnoldi method for solving the PEP (1.2). An approximation eigenpair (θ, z), where $\theta \in \mathscr{C}$ and $z \in \mathscr{Q}_m$, is obtained by imposing the Galerkin condition

$$(\theta^d A_d + \theta^{d-1}A_{d-1} + \cdots + \theta A_1 + A_0)z \perp \mathscr{Q}_m \, ; \tag{2.5}$$

and since z can be expressed as $z = Q_m\xi$ ($\xi \in \mathscr{C}^m$), this condition (2.5) may be rewritten as

$$(\theta^d \hat{A}_d + \theta^{d-1}\hat{A}_{d-1} + \cdots + \theta\hat{A}_1 + \hat{A}_0)\xi = 0 \, , \tag{2.6}$$

where

$$\hat{A}_i = Q_m^H A_i Q_m \quad (0 \le i \le d) \, . \tag{2.7}$$

The projected matrices $\hat{A}_i (0 \le i \le d)$ may therefore preserve some structures and properties as $A_i (0 \le i \le d)$ such as symmetry, and rhe projected polynomial eigenvalue problem (2.6) can be solved by the linearisation method.

However, as the dimension m of the projected subspace increases the generalised Arnoldi method becomes expensive and impractical due to storage requirements or computational cost, so restarting as in the explicitly restarted generalised Arnoldi (GAR) method [4] is desirable.

Algorithm 2.2. GAR(m): Explicitly restarted generalised Arnoldi method.
Input: Coefficient matrices $A_i (0 \le i \le d)$, initial vectors $u_i (0 \le i \le d-1)$, dimension m of the projected subspace and number k of the desired eigenpairs.
Output: k approximate eigenpairs and their relative residuals.

1. Run Algorithm 2.1 with $M_i = -A_d^{-1}A_i$ $(0 \le i \le d-1)$ to get Q_m;
2. Compute $\hat{A}_i$ $(0 \le i \le d)$ from (2.7);
3. Solve the projected polynomial eigenvalue problem (2.6) for (θ_i, ξ_i) $(1 \le i \le dm)$ with $|\theta_1| \ge \cdots \ge |\theta_{dm}|$, and get k Ritz pairs (θ_i, x_i) with $x_i = Q_m\xi_i / \|Q_m\xi_i\|_2$ $(1 \le i \le k)$;
4. Compute k relative residuals

$$\alpha_i = \frac{\|(\theta_i^d A_d + \cdots + \theta_i A_1 + A_0)x_i\|_2}{|\theta_i|^d\|A_d\|_F + \cdots + |\theta_i|\|A_1\|_F + \|A_0\|_F} \, (1 \le i \le k).$$

If all k relative residuals α_i are satisfied, then output (θ_i, x_i) and $\alpha_i (1 \le i \le k)$, and stop; otherwise set

$$\begin{pmatrix} u_{d-1} \\ u_{d-2} \\ \vdots \\ u_0 \end{pmatrix} = \sum_{i=1}^{k} \alpha_i \begin{pmatrix} \theta_i^{d-1} x_i \\ \theta_i^{d-2} x_i \\ \vdots \\ x_i \end{pmatrix}$$

as new initial vectors and go to 1.

The Ritz vectors may converge very slowly and even fail to converge, even if the Ritz values converge. To avoid this, Bao *et al.* [4] extended the refined projection principle [18] to the generalised Arnoldi method. Thus if $\theta_i (1 \le i \le k)$ are the desired Ritz values obtained by Algorithm 2.2, a refined Ritz vector $\hat{x}_i \in \mathscr{Q}_m$ corresponding to the Ritz value θ_i is sought such that

$$\hat{x}_i = \underset{x \in \mathscr{Q}_m, \|x\|_2 = 1}{\arg\min} \left\| (\theta_i^d A_d + \cdots + \theta_i A_1 + A_0) x \right\|_2 . \tag{2.8}$$

It was shown that $\hat{x}_i = Q_m \hat{\xi}_i$, where $\hat{\xi}_i$ is the right singular vector corresponding to the smallest singular value σ_i of the matrix

$$T_i = (\theta_i^d A_d + \cdots + \theta_i A_1 + A_0) Q_m \in \mathscr{C}^{n \times m} . \tag{2.9}$$

It is easy to see that the approximate eigenpair $(\theta_i, \hat{x}_i)$, called the refined Ritz pair, is better than the Ritz pair (θ_i, x_i) due to its minimal property. The corresponding relative residual α_i can be rewritten as

$$\alpha_i = \frac{\sigma_i}{|\theta_i|^d \|A_d\|_F + \cdots + |\theta_i| \|A_1\|_F + \|A_0\|_F} \ (1 \le i \le k) , \tag{2.10}$$

and the explicitly restarted refined generalised Arnoldi (RGAR) method [4] for solving the PEP is as follows.

Algorithm 2.3. RGAR(m): Explicitly restarted refined generalised Arnoldi method.
Input: Coefficient matrices $A_i (0 \le i \le d)$, initial vectors $u_i (0 \le i \le d-1)$, dimension m of the projected subspace and number k of the desired eigenpairs.
Output: k refined Ritz pairs and their relative residuals.

1. Implement Steps 1 and 2 in Algorithm 2.2;
2. Compute all the eigenvalues $\{\theta_i\}_{i=1}^{dm}$ of the projected polynomial eigenvalue problem (2.6) with $|\theta_1| \ge |\theta_2| \ge \cdots \ge |\theta_{dm}|$;
3. **for** $i = 1 : k$ **do**
   Compute the smallest singular value σ_i of T_i defined in (2.9) and the corresponding right singular vector $\hat{\xi}_i$, the refined Ritz vector $\hat{x}_i = Q_m \hat{\xi}_i$ and the corresponding relative residual (2.10);
   endfor

4. If all k relative residuals α_i are satisfied, then output $(\theta_i, \hat{x}_i)$ and $\alpha_i (1 \le i \le k)$, and stop; otherwise set

$$\begin{pmatrix} u_{d-1} \\ u_{d-2} \\ \vdots \\ u_0 \end{pmatrix} = \sum_{i=1}^{k} \alpha_i \begin{pmatrix} \theta_i^{d-1} \hat{x}_i \\ \theta_i^{d-2} \hat{x}_i \\ \vdots \\ \hat{x}_i \end{pmatrix}$$

as new initial vectors and go to 1.

Numerical results in [4] showed that the explicitly restarted refined generalised Arnoldi (RGAR) method is superior to the explicitly restarted generalised Arnoldi (GAR) method.

3. Implicitly Restarted Refined Generalised Arnoldi Method

We now develop our implicit restarting strategy for solving the PEP via the generalised Arnoldi method. Using Algorithm 2.1, we first obtain the decomposition (2.4). Thus for given $s = m-k$ shifts μ_i, we perform s implicitly shifted QR iterations [29] on H_m to obtain the relation

$$(H_m - \mu_1 I) \cdots (H_m - \mu_s I) = V_m R \,, \tag{3.1}$$

where $V_m = (v_{i,j}) \in \mathscr{C}^{m \times m}$ is a unitary matrix and $R \in \mathscr{C}^{m \times m}$ is an upper triangular matrix. It is straightforward to show that V_m has only s nonzero subdiagonals. Let

$$Q_m^+ = Q_m V_m = [Q_k^+, Q_s^+] = [q_1^+, q_2^+, \cdots, q_m^+] \,, \tag{3.2}$$

$$P_m^{(i)^+} = P_m^{(i)} V_m = [P_k^{(i)^+}, P_s^{(i)^+}] = [p_1^{(i)^+}, p_2^{(i)^+}, \cdots, p_m^{(i)^+}] \, (1 \le i \le d-1) \,, \tag{3.3}$$

$$H_m^+ = V_m^H H_m V_m = \begin{pmatrix} H_k^+ & H_{12}^+ \\ H_{21}^+ & H_{22}^+ \end{pmatrix}, \tag{3.4}$$

with $Q_k^+, P_k^{(i)^+} \in \mathscr{C}^{n \times k} (1 \le i \le d-1)$, $H_k^+ \in \mathscr{C}^{k \times k}$ and $q_j^+, p_j^{(i)^+} \in \mathscr{C}^n (1 \le j \le m, 1 \le i \le d-1)$. It follows from the relation (2.4) that

$$M \begin{pmatrix} Q_m^+ \\ P_m^{(1)^+} \\ \vdots \\ P_m^{(d-1)^+} \end{pmatrix} = \begin{pmatrix} Q_m^+ \\ P_m^{(1)^+} \\ \vdots \\ P_m^{(d-1)^+} \end{pmatrix} H_m^+ + h_{m+1,m} \begin{pmatrix} q_{m+1} \\ p_{m+1}^{(1)} \\ \vdots \\ p_{m+1}^{(d-1)} \end{pmatrix} e_m^T V_m \,, \tag{3.5}$$

where H_m^+ remains an upper Hessenberg matrix and Q_m^+ satisfies $(Q_m^+)^H Q_m^+ = I_m$. It is also notable that V_m has only s nonzero subdiagonals, and

$$e_m^T V_m = (0, \cdots, 0, v_{m,k}, \cdots, v_{m,m}) \,. \tag{3.6}$$

Since $H_m^+ = (h_{i,j}^+)$ is an upper Hessenberg matrix, the entry $h_{k+1,k}^+$ in the top right corner of H_{21}^+ may be nonzero, but the others are all zero. Now let

$$\begin{cases} \eta_k^+ = h_{k+1,k}^+ q_{k+1}^+ + h_{m+1,m} v_{m,k} q_{m+1} , \\ \tilde{h}_{k+1,k}^+ = \|\eta_k^+\|_2 , \\ q_{k+1}^+ = \eta_k^+ / \tilde{h}_{k+1,k}^+ , \\ p_{k+1}^{(i)^+} = (h_{k+1,k}^+ p_{k+1}^{(i)^+} + h_{m+1,m} v_{m,k} p_{m+1}^{(i)}) / \tilde{h}_{k+1,k}^+ , \ 1 \le i \le d-1 . \end{cases} \tag{3.7}$$

Comparing the first k columns of relation (3.5) and using Eqs. (3.2), (3.3) , (3.4) and (3.7), we obtain

$$M \begin{pmatrix} Q_k^+ \\ P_k^{(1)^+} \\ \vdots \\ P_k^{(d-1)^+} \end{pmatrix} = \begin{pmatrix} Q_k^+ \\ P_k^{(1)^+} \\ \vdots \\ P_k^{(d-1)^+} \end{pmatrix} H_k^+ + \tilde{h}_{k+1,k}^+ \begin{pmatrix} q_{k+1}^+ \\ p_{k+1}^{(1)^+} \\ \vdots \\ p_{k+1}^{(d-1)^+} \end{pmatrix} e_k^T . \tag{3.8}$$

It is easy to verify that $(Q_k^+)^H Q_k^+ = I_k, (Q_k^+)^H q_{k+1}^+ = 0$. As a truncated form of the generalised Arnoldi decomposition (2.4), this relation (3.8) indicates why in Section 5 our implicit method proves superior to the explicit method (Algorithm 2.2 or 2.3) that only uses new initial vectors, with the generalised Arnoldi procedure continued from the k-step to the m-step. The shifts selected are a key for success of the implicit restarting strategy. One way is to take the unwanted Ritz values as shifts, called exact shifts. We solve the projected polynomial eigenvalue problem (2.6) to get dm eigenvalues $\{\theta_i\}_{i=1}^{dm}$ with $|\theta_1| \ge |\theta_2| \ge \cdots \ge |\theta_{dm}|$, and select k Ritz values $\{\theta_i\}_{i=1}^{k}$ as approximations to the desired eigenvalues. The remaining unwanted Ritz values $\{\theta_i\}_{i=k+1}^{dm}$ are shift candidates. (There are $dm-k$ shift candidates, while we need only $s = m-k$ shifts.) In this case, we select the s Ritz values among $dm-k$ candidates $\{\theta_i\}_{i=k+1}^{dm}$ farthest from the Ritz values $\{\theta_i\}_{i=1}^{k}$ as shifts — i.e.

$$\mu_i = \theta_{(d-1)m+k+i} , \ 1 \le i \le s . \tag{3.9}$$

However, Jia & Sun [19] pointed out that this selection of exact shifts is problematic and susceptible to failure for the QEP , so they suggested using the refinement strategy to obtain better shifts (called refined shifts) for the implicit restarting process — and we extend this idea to the PEP .

The refined generalised Arnoldi method can not only improve the accuracy of the Ritz vectors but also provide more accurate approximations to some unwanted eigenvalues. If $\hat{\xi}_i$ denotes the right singular vector corresponding to the smallest singular value of the matrix T_i $(dm-s+1 \le i \le dm)$, then $\{\hat{x}_i = Q_m \hat{\xi}\}_{i=dm-s+1}^{dm}$ are the refined Ritz vectors corresponding to the unwanted Ritz values $\{\theta_i\}_{i=dm-s+1}^{dm}$. Using the refined Ritz vector $\hat{x}_i$ $(dm-s+1 \le i \le dm)$, we can obtain a more accurate approximate eigenvalue θ by imposing the Galerkin condition

$$(\theta^d A_d + \theta^{d-1} A_{d-1} + \cdots + \theta A_1 + A_0)\hat{x}_i \perp \hat{x}_i . \tag{3.10}$$

The condition (3.10) is equivalent to the polynomial scalar equation

$$a_d^{(i)}\theta^d + \cdots + a_1^{(i)}\theta + a_0^{(i)} = 0\,, \tag{3.11}$$

where $a_j^{(i)} = \hat{\xi}_i^H \hat{A}_j \hat{\xi}_i$ $(0 \le j \le d)$, and $\hat{A}_j$ is defined by (2.7). From Eq. (3.11) we obtain d roots $\{\omega_j^{(i)}\}_{j=1}^d$ with $|\omega_1^{(i)}| \le |\omega_2^{(i)}| \le \cdots \le |\omega_d^{(i)}|$. Using the s refined Ritz vectors $\{\hat{x}_i\}_{i=dm-s+1}^{dm}$, we get sd roots $\{\omega_j^{(i)}\}_{j=1}^d$ $(dm-s+1 \le i \le dm)$. Then we select the s values among sd candidates $\{\omega_j^{(i)}\}$ $(1 \le j \le d; dm-s+1 \le i \le dm)$ farthest from the Ritz values $\{\theta_i\}_{i=1}^k$ as shifts (the refined shifts). Now we propose our implicitly restarted refined generalised Arnoldi (IRGAR) method for solving the PEP as follows.

Algorithm 3.1. IRGAR(m): Implicitly restarted refined generalised Arnoldi method.
Input: Coefficient matrices $A_i (0 \le i \le d)$, initial vectors $u_i (0 \le i \le d-1)$, dimension m of the projected subspace and number k of the desired eigenpairs.
Output: k approximate eigenpairs and their relative residuals.

1. Implement Steps 1, 2 and 3 in Algorithm 2.3;
2. If all k relative residuals α_i are satisfied, then output $(\theta_i, \hat{x}_i)$ and $\alpha_i (1 \le i \le k)$, and stop; else set $s := m-k$. Go to 3 if refined shifts are picked; else go to 4;
3. **for** $i = dm-s+1 : dm$ **do**
   Compute the right singular vector $\hat{\xi}_i$ corresponding to the smallest singular value of T_i defined in (2.9), and solve the polynomial scalar equation (3.11) to get its roots $\{\omega_j^{(i)}\}_{j=1}^d$;
   endfor
4. Select s values among $\{\theta_i\}$ $(k+1 \le i \le dm)$ or $\{\omega_j^{(i)}\}$ $(1 \le j \le d; dm-s+1 \le i \le dm)$ farthest from the Ritz values $\{\theta_i\}_{i=1}^k$ as shifts $\mu_i (1 \le i \le s)$, and perform s implicit shifted QR iteration as (3.1), and compute the new matrices $Q_k^+, P_k^{(i)^+} (1 \le i \le d-1)$, vectors $q_{k+1}^+, p_{k+1}^{(i)^+} (1 \le i \le d-1)$ and scalar $\tilde{h}_{k+1,k}^+$ as (3.2), (3.3), (3.4) and (3.7), then go to 1.

In general, the IRGAR method with refined shifts (abbreviated as IRGAR with RS) consumes much more CPU time than the IRGAR method with exact shifts (abbreviated as IRGAR with ES), since it involves computing the refined approximate eigenvectors and the roots of the polynomial scalar equation (3.11). A numerical comparison of results obtained using exact shifts and refined shifts is presented in Section 5.

4. Deflated and Implicitly Restarted Refined Generalised Arnoldi Method

In the spirit of the non-equivalence low-rank deflation technique [33], we develop a novel non-equivalence low-rank deflation technique for the PEP and incorporate this

with the implicitly restarted refined generalised Arnoldi method. Suppose r eigenpairs (λ_i, x_i, y_i) $(1 \le i \le r)$ of the PEP (1.2) have already been obtained, and let

$$\Lambda_1 = diag_r(\lambda_1, \lambda_2, \cdots, \lambda_r), \quad X_1 = [x_1, x_2, \cdots, x_r], \quad Y_1 = [y_1, y_2, \cdots, y_r]$$

such that Λ_1, X_1 and Y_1 satisfy

$$P(\Lambda_1)X_1 = 0,\ Y_1^H P(\Lambda_1) = 0\,. \tag{4.1}$$

Assuming the left eigenvectors $y_1, y_2, \ldots, y_r$ are chosen such that $Y_1^H X_1 = I_r$, we construct a new matrix polynomial

$$\tilde{P}(\lambda) = \lambda^d \tilde{A}_d + \lambda^{d-1}\tilde{A}_{d-1} + \cdots + \lambda\tilde{A}_1 + \tilde{A}_0\,, \tag{4.2}$$

involving the new coefficient matrices

$$\begin{cases} \tilde{A}_0 = A_0 + A_1X_1\Lambda_1Y_1^H + A_2X_1\Lambda_1^2Y_1^H + \cdots + A_dX_1\Lambda_1^dY_1^H\,, \\ \tilde{A}_1 = A_1 + A_2X_1\Lambda_1Y_1^H + A_3X_1\Lambda_1^2Y_1^H + \cdots + A_dX_1\Lambda_1^{d-1}Y_1^H\,, \\ \tilde{A}_2 = A_2 + A_3X_1\Lambda_1Y_1^H + A_4X_1\Lambda_1^2Y_1^H + \cdots + A_dX_1\Lambda_1^{d-2}Y_1^H\,, \\ \vdots \\ \tilde{A}_{d-1} = A_{d-1} + A_dX_1\Lambda_1Y_1^H\,, \\ \tilde{A}_d = A_d. \end{cases} \tag{4.3}$$

The matrix polynomials (1.1) and (4.2) are then related as follows.

Theorem 4.1. *Assume that the diagonal matrix $\Lambda_1 \in \mathscr{C}^{r\times r}$ and the matrices $X_1, Y_1 \in \mathscr{C}^{n\times r}$ satisfy* (4.1) *and $Y_1^H X_1 = I_r$, and the new matrix polynomial $\tilde{P}(\lambda)$ is as in Eq.* (4.2)*. Then*

$$\tilde{P}(\lambda) = P(\lambda)\big[I_n + X_1\Lambda_1(\lambda I_r - \Lambda_1)^{-1}Y_1^H\big] \tag{4.4}$$

and the eigenvalues of $\tilde{P}(\lambda)$ are those of $P(\lambda)$, except that eigenvalues of Λ_1 are replaced by r zeros.

Proof. It follows from Eq. (4.2) and (4.3) that

$$\begin{aligned} \tilde{P}(\lambda) =& \lambda^d\tilde{A}_d + \lambda^{d-1}\tilde{A}_{d-1} + \lambda^{d-2}\tilde{A}_{d-2} + \cdots + \lambda\tilde{A}_1 + \tilde{A}_0 \\ =& \lambda^d A_d + \lambda^{d-1}(A_{d-1} + A_dX_1\Lambda_1Y_1^H) + \lambda^{d-2}(A_{d-2} + A_{d-1}X_1\Lambda_1Y_1^H + A_dX_1\Lambda_1^2Y_1^H) \\ &+ \cdots + \lambda(A_1 + A_2X_1\Lambda_1Y_1^H + \cdots + A_{d-1}X_1\Lambda_1^{d-2}Y_1^H + A_dX_1\Lambda_1^{d-1}Y_1^H) \\ &+ (A_0 + A_1X_1\Lambda_1Y_1^H + \cdots + A_{d-1}X_1\Lambda_1^{d-1}Y_1^H + A_dX_1\Lambda_1^dY_1^H) \\ =& P(\lambda) + A_dX_1(\lambda^{d-1}I_r + \lambda^{d-2}\Lambda_1 + \cdots + \lambda\Lambda_1^{d-2} + \Lambda_1^{d-1})\Lambda_1Y_1^H \\ &+ A_{d-1}X_1(\lambda^{d-2}I_r + \lambda^{d-3}\Lambda_1 + \cdots + \lambda\Lambda_1^{d-3} + \Lambda_1^{d-2})\Lambda_1Y_1^H \\ &+ \cdots + A_2X_1(\lambda I_r + \Lambda_1)\Lambda_1Y_1^H + A_1X_1\Lambda_1Y_1^H \end{aligned}$$

so that

$$\begin{aligned}\tilde{P}(\lambda) =& P(\lambda) + \big[A_d X_1(\lambda^d I_r - \Lambda_1^d) + A_{d-1}X_1(\lambda^{d-1}I_r - \Lambda_1^{d-1}) \\ &+ \cdots + A_2 X_1(\lambda^2 I_r - \Lambda_1^2) + A_1 X_1(\lambda I_r - \Lambda_1)\big](\lambda I_r - \Lambda_1)^{-1}\Lambda_1 Y_1^H \\ =& P(\lambda) + (\lambda^d A_d + \lambda^{d-1}A_{d-1} + \cdots + \lambda A_1 + A_0)X_1(\lambda I_r - \Lambda_1)^{-1}\Lambda_1 Y_1^H \\ =& P(\lambda) + P(\lambda)X_1(\lambda I_r - \Lambda_1)^{-1}\Lambda_1 Y_1^H \\ =& P(\lambda)\big[I_n + X_1\Lambda_1(\lambda I_r - \Lambda_1)^{-1}Y_1^H\big] .\end{aligned}$$

From Eq. (4.4) and the identity $det(I_n + RS) = det(I_m + SR)$ where $R \in \mathscr{C}^{n\times m}$ and $S \in \mathscr{C}^{m\times n}$, we have

$$det(\tilde{P}(\lambda)) = det(P(\lambda))det\big(I_r + \Lambda_1(\lambda I_r - \Lambda_1)^{-1}\big) = \frac{\lambda^r det(P(\lambda))}{det(\lambda I_r - \Lambda_1)} .$$

Hence $\tilde{P}(\lambda)$ has the same eigenvalues as $P(\lambda)$ except that r eigenvalues of Λ_1 are replaced by r zeros. □

Using Theorem 4.1, we can transform the converged eigenvalues of the PEP (1.2) to zeros so that the next desired eigenvalues become the eigenvalues with the largest modulus. Suppose that $(\lambda_1, x_1, y_1) \in \mathscr{C} \times \mathscr{C}^n \times \mathscr{C}^n$ is a converged eigenpair of the PEP (1.2) such that $y_1^H x_1 = 1$, when Eqs. (4.3) reduce to

$$\begin{cases} \tilde{A}_i = A_i + \sum\limits_{j=i+1}^{d} \lambda_1^{j-i} A_j x_1 y_1^H , \ 0 \le i \le d-1 , \\ \tilde{A}_d = A_d . \end{cases} \tag{4.5}$$

If the non-equivalence low-rank deflation technique has been performed, then the relation (3.8) no longer holds with the new coefficient matrices $\tilde{A}_i (0 \le i \le d)$. For simplicity, we restart the generalised Arnoldi method on taking

$$u_{d-1} = q_1^+, \quad u_i = p_1^{(d-i-1)^+} \quad (0 \le i \le d-2) \tag{4.6}$$

as the new initial vectors, and our deflated and implicitly restarted refined generalised Arnoldi method (DIRGAR) for solving the PEP is as follows.

Algorithm 4.1. DIRGAR(m): Deflated and implicitly restarted refined generalised Arnoldi method.
Input: Coefficient matrices $A_i (0 \le i \le d)$, initial vectors $u_i (0 \le i \le d-1)$, dimension m of the projected subspace and number k of the desired eigenpairs.
Output: k approximate eigenpairs and their relative residuals.

1. Implement Steps 1, 2 and 3 in Algorithm 2.3;

2. Set $Idef = 0$. If all k relative residuals α_i are satisfied, then output $(\theta_i, \hat{x}_i)$ and $\alpha_i (1 \le i \le k)$, and stop; if none of α_i is satisfied, then set $s := m - k$ and go to

3 if refined shifts are picked; else go to 4; if r $(0 < r < k)$ relative residuals α_i are satisfied, then output the r converged eigenpairs and the corresponding relative residuals, and set $k := k-r, s := m-k, Idef = 1$. Go to 3 if refined shifts are picked; else go to 4;

3. **for** $i = dm-s+1 : dm$ **do**
Compute the eigenvector $\hat{\xi}_i$ corresponding to the smallest eigenvalue of $T_i^H T_i$, and solve the polynomial scalar equation (3.11) to get its roots $\{\omega_j^{(i)}\}_{j=1}^d$;
endfor

4. Select s values among $\{\theta_i\}$ $(k+1 \le i \le dm)$ or $\{\omega_j^{(i)}\}$ $(1 \le j \le d; dm-s+1 \le i \le dm)$ farthest from the Ritz values $\{\theta_i\}_{i=1}^k$ as shifts $\mu_i (1 \le i \le s)$, and perform s implicit shifted QR iteration as (3.1), and compute the new matrices $Q_k^+, P_k^{(i)^+} (1 \le i \le d-1)$, vectors $q_{k+1}^+, p_{k+1}^{(i)^+} (1 \le i \le d-1)$ and scalar $\tilde{h}_{k+1,k}^+$ as (3.2), (3.3), (3.4) and (3.7);

5. If $Idef = 0$, then go to 1; otherwise go to 6;

6. Compute r left eigenvectors of the PEP (1.2) corresponding to the r converged eigenvalues with $y_i^H x_i = 1$ $(1 \le i \le r)$;

7. Perform the non-equivalence low-rank deflation via (4.5) for each converged eigenpair, update the new initial vectors as (4.6), and go to 1.

5. Numerical Experiments

In this section, we report some numerical examples to illustrate the effectiveness of the IRGAR and DIRGAR methods, and compare them both with the RGAR method. All numerical calculations were performed in MATLAB R2015b on an Intel Core 2.9 GHz PC with 4GB memory under the Windows 7 system. The initial vectors are all randomly chosen, n and m denote the respective dimensions of the PEP (1.2) and the projected subspace, k denotes the number of desired eigenpairs with the largest modulus eigenvalues, $Iter$ and $Iter_{max} = 500$ denote the number and maximum number of restarting steps, CPU is the CPU time (in seconds) for implementing the relevant algorithm, and the tolerance $tol = 10^{-10}$ is adopted in the stopping criterion for relative residuals $Res = \max_i\{\alpha_i\} \le tol$.

Example 5.1. Let us consider the cubic eigenvalue problem

$$P(\lambda)x = (\lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0)x = 0\,, \tag{5.1}$$

where the coefficient matrices arise in the “plasma-drift" problem in Ref. [5] and the matrices are 512×512. Using the MATLAB function $polyeig(4)$ or $eigs(4)$, we obtained the

Table 1: Numerical results for Example 5.1.

Methods	$Iter$	CPU	α_i	computed eigenvalues
RGAR(20)	25	0.67	0.8669e-10	47.70640814694 - 0.00678490010i
			0.4807e-10	-47.57496119269 - 0.00669146280i
			0.8583e-10	47.09896131054 - 0.00678654473i
			0.8274e-10	-46.96756223198 - 0.00669187989i
IRGAR(20) with ES	8	0.31	0.8539e-10	47.70640814711 - 0.00678489997i
			0.4644e-10	-47.57496119288 - 0.00669146293i
			0.5154e-10	47.09896131016 - 0.00678654527i
			0.5299e-10	-46.96756223153 - 0.00669187937i
IRGAR(20) with RS	7	0.30	0.8153e-10	47.70640814694 - 0.00678490012i
			0.3226e-10	-47.57496119273 - 0.00669146278i
			0.3497e-10	47.09896131073 - 0.00678654451i
			0.6009e-10	-46.96756223208 - 0.00669188013i

four largest modulus eigenvalues

$$\begin{aligned}
\lambda_1 &= 47.706408145293460 - 0.006784904974176i,\\
\lambda_2 &= -47.574961194358565 - 0.006691467596723i,\\
\lambda_3 &= 47.098961311207900 - 0.006786543786995i,\\
\lambda_4 &= -46.967562232594950 - 0.006691880875913i.
\end{aligned}$$

Setting $m = 20$ and $k = 4$, we used the RGAR method, the IRGAR with ES and then the IRGAR with RS, to compute 4 eigenpairs with largest modulus eigenvalues shown in Table 1.

All three methods converge, and the IRGAR(20) with ES and IRGAR(20) with RS are both seen to be superior to the RGAR(20) in the number of restarting process and the CPU time, so the implicitly restarting process improves the RGAR method. We also observe that the number of restarting steps and the CPU time in the IRGAR(20) with RS are slightly less than in the IRGAR(20) with ES.

Example 5.2. Let us consider the quartic eigenvalue problem

$$P(\lambda)x = (\lambda^4 A_4 + \lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0)x = 0\,, \tag{5.2}$$

where the coefficient matrices come from the “planar-waveguide" problem in Ref. [5] and the matrices are 129×129. Setting $m = 20$, we used the RGAR method, the IRGAR with ES and the IRGAR with RS to compute k eigenpairs with the largest modulus eigenvalues, where the parameter k was chosen to be 4, 6 and 8, respectively. The numerical results are shown in Table 2.

Both the IRGAR(20) with ES and IRGAR(20) with RS are seen to converge but the RGAR(20) does not, even if the number of restarting process reaches the maximum number

Table 2: Numerical results for Example 5.2.

k	RGAR			IRGAR with ES			IRGAR with RS		
	$Iter$	CPU	Res	$Iter$	CPU	Res	$Iter$	CPU	Res
4	500	10.53	0.4493e-03	102	3.87	0.9173e-10	102	5.21	0.9865e-10
6	500	10.81	0.6063e-03	100	3.73	0.8507e-10	99	4.78	0.8752e-10
8	500	10.75	0.5514e-03	130	4.43	0.9220e-10	130	5.68	0.9546e-10

$Iter_{max}$. We also observe that the numbers of restarting steps in the IRGAR(20) with RS are equal to or slightly less than those in the IRGAR(20) with ES, but the IRGAR(20) with RS consumes some CPU time to compute the refined shifts.

Example 5.3 (cf. Ref. [4])**.** Let us now consider the cubic eigenvalue problem (5.1) when

$$A_3 = 5I_n,\ A_2 = \begin{pmatrix} 9 & -3 & & \\ -3 & 9 & \ddots & \\ & \ddots & \ddots & -3 \\ & & -3 & 9 \end{pmatrix} \in \mathscr{R}^{n\times n},$$

A_1 and A_0 come from the Harwell-Boeing test matrix bwm200 in Ref. [8] and the matrices are 200×200. Using the MATLAB function $polyeig(20)$ or $eigs(20)$, we found the following 20 largest modulus eigenvalues:

$$\begin{aligned}
\lambda_1 &= -16.818263252077116, & \lambda_2 &= -16.811593572838266,\\
\lambda_3 &= -16.800480319289290, & \lambda_4 &= -16.784926442755477,\\
\lambda_5 &= -16.764938899706028, & \lambda_6 &= -16.740521953786430,\\
\lambda_7 &= -16.711687826718137, & \lambda_8 &= -16.678441095233900,\\
\lambda_9 &= -16.640800204192725, & \lambda_{10} &= -16.598769047534688,\\
\lambda_{11} &= -16.552373231465260, & \lambda_{12} &= -16.501614978427103,\\
\lambda_{13} &= -16.446527970859920, & \lambda_{14} &= -16.387111781953113,\\
\lambda_{15} &= -16.323409058429892, & \lambda_{16} &= -16.255415761096930,\\
\lambda_{17} &= -16.183184358833053, & \lambda_{18} &= -16.106706255110325,\\
\lambda_{19} &= -16.026044565479502, & \lambda_{20} &= -15.941185212768882.
\end{aligned}$$

Setting $m = 30$ and $k = 20$, we used the RGAR method, the IRGAR method with exact shifts and the DIRGAR method with exact shifts to compute 20 eigenpairs with the largest modulus eigenvalues. Fig. 1 shows the maximum relative residuals of the 20 desired eigenpairs computed by the RGAR(30), IRGAR(30) and the DIRGAR(30). We observe that only the DIRGAR(30) converges, while the maximum relative residuals computed from the RGAR(30) and the IRGAR(30) oscillate around 10^{-3}. The deflation history of DIRGAR(30) in Table 3 shows the deflation process is necessary when more eigenpairs are sought, and the DIRGAR(30) is superior to both the RGAR(30) and IRGAR(30).

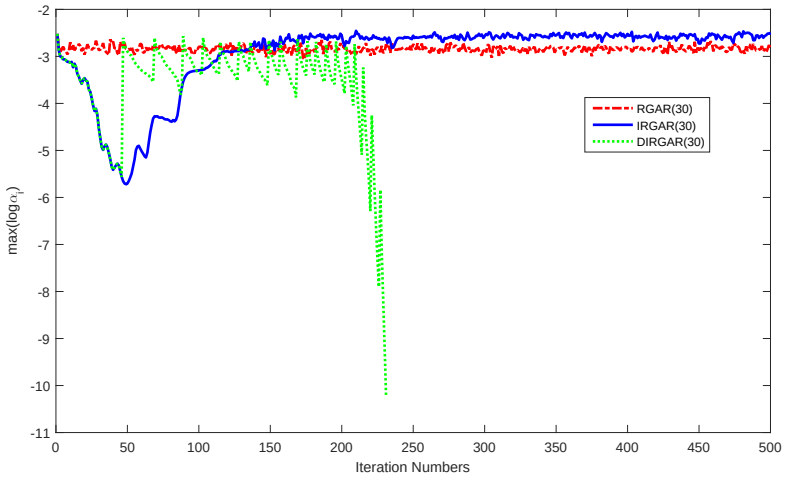

Figure 1: Convergence history of RGAR(30), IRGAR(30), DIRGAR(30) for Example 5.3.

Table 3: Deflation history of DIRGAR(30) for Example 5.3.

computed eigenvalues	α_i	deflation process
-16.81826325207	0.8175e-10	the 1st deflation
-16.81159357284	0.5762e-10	the 2nd deflation
-16.80048031929	0.3086e-10	the 3rd deflation
-16.78492644275	0.9765e-10	the 4th deflation
-16.76493889969	0.9386e-10	the 5th deflation
-16.74052195379	0.6242e-10	the 6th deflation
-16.71168782671	0.5285e-10	the 7th deflation
-16.67844109523	0.7992e-10	the 8th deflation
-16.64080020420	0.7870e-10	the 9th deflation
-16.59876904752	0.9753e-10	the 10th deflation
-16.55237323147	0.4537e-10	the 11th deflation
-16.50161497844	0.3745e-10	the 12th deflation
-16.44652797083	0.9818e-10	the 13th deflation
-16.38711178192	0.9653e-10	the 14th deflation
-16.32340905843	0.2909e-10	the 15th deflation
-16.25541576110	0.7592e-10	the 16th deflation
-16.18318435884	0.8878e-10	the 17th deflation
-16.10670625510	0.4921e-10	the 18th deflation
-16.02604456549	0.7263e-10	the 19th deflation
-15.94118521279	0.3999e-10	the 20th deflation

In summary, our numerical results show that while the RGAR and IRGAR methods can compute a few eigenpairs with the largest modulus eigenvalues, the DIRGAR method may compute more eigenpairs for the polynomial eigenvalue problem.

6. Conclusion

Based on implicitly shifted QR iteration, we have developed an implicitly restarted version of the generalised Arnoldi method to solve the polynomial eigenvalue problem. The implicitly restarted strategy combined with the refinement scheme provides an implicitly restarted refined generalised Arnoldi method for computing a few eigenpairs for the polynomial eigenvalue problem. However, a novel explicit non-equivalence low-rank deflation technique for the polynomial eigenvalue problem led to a deflated and implicitly restarted refined generalised Arnoldi method to compute more eigenpairs, and some numerical results indicate that this method is efficient and robust.

Acknowledgments

The authors thank Professor Roger J. Hosking and two anonymous referees for valuable comments and suggestions, which helped to improve the presentation of this work. The research is supported by the National Natural Science Foundation of China under grant No.11571171.

References

[1] B. Adhikari, R. Alam and D. Kressner, *Structured eigenvalue condition numbers and linearizations for matrix polynomials*, Linear Algebra Appl. **435**, 2193-2221 (2011).
[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H.A. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problem: A Practical Guide*, SIAM, Philadelphia (2000).
[3] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl. **26**, 640-659 (2005).
[4] L. Bao, Y. Lin and Y. Wei, *Restarted generalized Krylov subspace methods for solving large-scale polynomial eigenvalue problems*, Numer. Algor. **50**, 17-32 (2009).
[5] T. Betcke, N.J. Higham, V. Mehrmann, C. Schroder and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, ACM Trans. Math. Soft. **39**, 7-28 (2013).
[6] E.K.-W. Chu, *Perturbation of eigenvalues for matrix polynomials via the Bauer-Fike theorems*, SIAM J. Matrix Anal. Appl. **25**, 551-573 (2003).
[7] J. Dedieu and F. Tisseur, *Perturbation theory for homogeneous polynomial eigenvalue problems*, Linear Algebra Appl. **358**, 71-94 (2003).
[8] I.S. Duff, R.G. Grimes and J.G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Soft. **15**, 1-14 (1989).
[9] I. Gohberg, P. Lancaster and L. Rodman, *Matrix Polynomials*, Academic Press, New York (1982).
[10] N.J. Higham, R.-C. Li and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, SIAM J. Matrix Anal. Appl. **29**, 1218-1241 (2007).

[11] N.J. Higham, D.S. Mackey and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl. **28**, 1005-1028 (2006).
[12] N.J. Higham and F. Tisseur, *More on pseudo spectra for polynomial eigenvalue problems and applications in control theory*, Linear Algebra Appl. **351**, 435-453 (2002).
[13] N.J. Higham and F. Tisseur, *Bounds for eigenvalues of matrix polynomials*, Linear Algebra Appl. **358**, 5-22 (2003).
[14] W.-Q. Huang, T.-X. Li, Y.-T. Li and W.-W. Lin, *A semiorthogonal generalized Arnoldi method and its variations for quadratic eigenvalue problems*, Numer. Linear Algebra Appl. **20**, 259-280 (2013).
[15] T.-M. Hwang, W.-W. Lin, J.-L. Liu and W. Wang, *Jacobi-Davidson methods for cubic eigenvalue problems*, Numer. Linear Algebra Appl. **12**, 605-624 (2005).
[16] T.-M. Hwang, W.-W. Lin, W.-C. Wang and W. Wang, *Numerical simulation of three dimensional pyramid quantum dot*, J. Comput. Phys. **196**, 208-232 (2004).
[17] F.-N. Hwang, Z.-H. Wei, T.-M. Hwang and W. Wang, *A parallel additive Schwarz preconditioned Jacobi-Davidson algorithm for polynomial eigenvalue problems in quantum dot simulation*, J. Comput. Phys. **229**, 2932-2947 (2010).
[18] Z.-X. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl. **259**, 1-23 (1997).
[19] Z.-X. Jia and Y.-Q. Sun, *Implicitly restarted generalized second-order Arnoldi type algorithms for the quadratic eigenvalue problem*, Taiwanese J. Math. **19**, 1-30 (2015).
[20] D. Kressner and J.E. Roman, *Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis*, Numer. Linear Algebra Appl. **21**, 569-588 (2014).
[21] P.W. Lawrence and R.M. Corless, *Backward error of polynomial eigenvalue problems solved by linearization of Lagrange interpolants*, SIAM J. Matrix Anal. Appl. **36**, 1425-1442 (2015).
[22] D. Lu, Y. Su and Z. Bai, *Stability analysis of the two-level orthogonal Arnoldi procedure*, SIAM J. Matrix Anal. Appl. **37**, 195-214 (2016).
[23] D.S. Mackey, N. Mackey, C. Mehl and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl. **28**, 971-1004 (2006).
[24] D.S. Mackey, N. Mackey, C. Mehl and V. Mehrmann, *Structured polynomial eigenvalue problems: good vibrations from good linearization*, SIAM J. Matrix Anal. Appl. **28**, 1029-1051 (2006).
[25] K. Meerbergen, *The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl. **30**, 1463-1482 (2008).
[26] J.S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill, New York (1968).
[27] K. Rothe, *Least squares element method for boundary eigenvalue problems*, Int. J. Numer. Meth. Engng. **33**, 2129-2143 (1992).
[28] G.L.G. Sleijpen, A.G.L. Booten, D.R. Fokkema and H.A. van der Vorst, *Jacobi-Davidson type methods for generalised eigenproblems and polynomial eigenproblems*, BIT **36**, 595-633 (1996).
[29] D.C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl. **13**, 357-385 (1992).
[30] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl. **309**, 339-361 (2000).
[31] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev. **43**, 235-286 (2001).
[32] W. Wang, T.-M. Hwang, W.-W. Lin and J.-L. Liu, *Numerical methods for semiconductor heterostructures with band nonparabolicity*, J. Comput. Phys. **190**, 141-158 (2003).
[33] W. Wei and H. Dai, *Implicitly restarted refined partially orthogonal projection method with deflation*, East Asian J. Appl. Math. **7**, 1-20 (2017).