EVALUATION OF ACCELERATION TECHNIQUES FOR THE RESTARTED ARNOLDI METHOD

西田 晃　小柳 義夫

AKIRA NISHIDA* AND YOSHIO OYANAGI*

Abstract. We present an approach for the acceleration of the restarted Arnoldi iteration for the computation of a number of eigenvalues of the standard eigenproblem $Ax = \lambda x$. This study applies the Chebyshev polynomial to the restarted Arnoldi iteration and proves that it computes necessary eigenvalues with far less complexity than the QR method. We also discuss the dependence of the convergence rate of the restarted Arnoldi iteration on the distribution of spectrum. This research aims to compare this algorithm with other state-of-the-art approaches.

Key words. sparse nonsymmetric eigenvalue problems, restarted Arnoldi iteration, polynomial acceleration

1. Introduction. The situation of the computation of eigenpairs for the standard eigenproblem was not satisfactory, since they can not be treated by the direct approaches. The QR method requires $\mathcal{O}(n^3)$ arithmetic operations, which puts restrictions on n. For nonsymmetric matrices, Arnoldi's method [1], the two-sided Lanczos method [2], and Davidson's method [5] were available, although their behavior was still less understood.

In the past five years, there have been great progress in the further developments of these methods. Arnoldi's method, which had the defect of increasing computational complexity per iteration step, was much improved by Saad [15] with the explicitly restarting technique, by which the dimensions of the searchspaces can be kept modest. Although the restarted Arnoldi iteration is quite effective, the dimension of the subspace is inevitably large, in particular when the wanted eigenvalues are clustered. Moreover it favors the convergence on the envelope of the spectrum. In this paper, we use the convex hull proposed for the solution of the nonsymmetric linear system to accelerate the convergence of the restarted Arnoldi iteration. We also mention the relatively recent variant developed by Sorensen [20], the implicitly restarted Arnoldi iteration, which is a truncation of the standard implicitly shifted Arnoldi iteration.

2. Background. We will give an outline of the methods referred to in this paper here. The idea of the iteration techniques is explained briefly. We then describe the least-squares based method, which were originally developed for solving the linear system [16][17].

2.1. The Arnoldi iteration. The Arnoldi approach involves the column-by-column generation of an orthogonal Q such that $Q^T AQ = H$ is the Hessenberg reduction[7]. If $Q = [q_1, ..., q_l]$ and we isolate the last term in the summation $Aq_l = \sum_{i=1}^{l+1} h_{il} q_l$, then

$$h_{l+1,l} q_{l+1} = Aq_l - \sum_{i=1}^{l} h_{il} q_l :\equiv r_l$$

* Department of Information Science, Faculty of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113 Japan (nishida@is.s.u-tokyo.ac.jp).

where $h_{il} = q_i^T A q_l$ for $i = 1, ..., l$. We assume that q_1 is a given 2-norm starting vector. The Arnoldi process computes an orthonormal basis for the Krylov subspace $\mathcal{K}(A, q_1, l)$:

$$\text{span}\{q_1, ..., q_l\} = \text{span}\{q_1, Aq_1, ..., A^{l-1}q_1\},$$

in which the map is represented by an upper Hessenberg matrix H_l.

1. $h_{1,1} = (Aq_1, q_1)$;
2. for $j = 1, ..., l-1$, put

$$r_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i, \quad h_{j+1,j} = \| r_j \|_2,$$

$$q_{j+1} = h_{j+1}^{-1} r_j, \quad h_{i,j+1} = (Aq_{j+1}, q_i), \quad (i \le j+1).$$

The situation after l steps is summarized by the l-step Arnoldi factorization

$$AQ_l = Q_l H_l + r_l e_k^T$$

where $e_k = (0, ..., 0, 1)^T$ and $H_l = (h_{ij})$. The algorithm terminates when $r_j = 0$, which is impossible if the minimal polynomial of A with respect to q_1 is of degree $\ge l$. If this condition is satisfied, H_l is an irreducible Hessenberg matrix.

In the iterative variant [16], we start with an initial vector q_1 and fix a moderate value m, then compute the eigenvectors of H_m. We begin again, using as a starting vector a linear combination of the eigenvectors.

2.2. Explicitly restarted Arnoldi iteration. The algorithm of the explicitly restarted Arnoldi iteration is summarized in Table 1. The choice of m is usually a tradeoff between the length of the reduction that may be tolerated and the rate of convergence. The accuracy of the Ritz values typically increases as m does. For most problems, the size of m is determined experimentally.

TABLE 1
A block version of explicitly restarted Arnoldi reduction with polynomial acceleration

1. Choose $V_1 \in \mathrm{R}^{n \times r}$.
2. For $j = 1, ..., m-1$ do
   $W_j = AV_j$
   For $i = 1, ..., j$ do
   $H_{i,j} = V_i^T W_j$; $W_j = W_j - V_i H_{i,j}$
   end for
   $Q_j R_j = W_j$; $V_{j+1} = Q_j$; $H_{j+1,j} = R_j$
   end for
3. Compute the eigenvalues of $H_m = (H_{i,j}) \in \mathrm{R}^{mr \times mr}$ and select $\{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$ of largest real parts.
4. Stop if their Ritz vectors $\tilde{X}_0 = \{\tilde{x}_1, ..., \tilde{x}_r\}$ satisfy the convergence criteria.
5. Define the iteration polynomial $p_k(\lambda)$ of degree k by $\text{Sp}(H_m) - \{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$.
6. $\tilde{X}_k = p_k(A)\tilde{X}_0$; $Q_k R_k = \tilde{X}_k$; $V_1 = Q_k$
7. Goto 2.

2.3. Implicitly restarted Arnoldi iteration. The ARPACK software package [9] implements an implicitly restarted Arnoldi method. The scheme is called implicit because the starting vector is updated with an implicitly shifted QR algorithm on the Hessenberg matrix H_m. This method is motivated by the following result:

Let $AX_m = X_m H_m + r_m e_m^T$ be a length m Arnoldi reduction and $\phi(\cdot)$ a polynomial of degree $p = m - k$ where $k < m$. Since

$$\phi(A)X_k = X_m \phi(H_m)[e_1 \ e_2 \ \cdots \ e_k]$$

holds, if we compute the QR factorization of $\phi(H_m)[e_1 \ e_2 \ \cdots \ e_k] = Q_k R_k$ then the columns of $X_m Q_k$ are an orthogonal basis for $\mathcal{R}(\phi(A)X_k)$, where we denote by $\mathcal{R}(A)$ the range of matrix A.

Table 2 gives the basic algorithm as implemented by ARPACK. Note that the convergence rate of the method does not depend on the distribution of the spectrum.

TABLE 2
An implicitly restarted Arnoldi iteration as implemented by `ARPACK`

- *Start*: Build a length m Arnoldi reduction $AX_m = X_m H_m + r_m e_m^T$ with the starting vector x_1.
- *Iteration*: Until convergence
  1. Compute the eigensystem $H_m S_m = S_m D_m$ ordered with the k wanted eigenvalues located in the leading portion of the quasi-diagonal matrix D_m.
  2. Perform $m - k = p$ steps of the QR iteration with the unwanted eigenvalues of D_m as shifts to obtain $H_m Q_m = Q_m H_m^+$.
  3. *Restart*: Postmultiply the length m Arnoldi reduction with Q_k to obtain the length k Arnoldi reduction $AX_m Q_k = X + mQ_k H_k^+ + r_k^+ e_k^T$. Q_k represents the matrix consisting of the leading k columns of Q_m, and H_k^+ is the leading principal submatrix of order k of H_m^+.
  4. Extend the length k Arnoldi reduction to a length m one.

3. Polynomial Accelerations Techniques. Suppose A is diagonalizable with eigenpairs (u_j, λ_j) for $j = 1, ..., n$. If $\psi(\cdot)$ is some polynomial and we expand the current starting vector x_1 in terms of the basis of eigenvectors, then

$$\psi(A)x_1 = u_1\psi(\lambda_1)\zeta_1 + \cdots + u_n\psi(\lambda_n)\zeta_n$$

Assuming that the eigenpairs (u_i, λ_i) are ordered so that the wanted k ones are at the beginning of the expansion, we seek a polynomial such that

$$\max_{i=k+1,...,n} |\psi(\lambda_i)| < \min_{i=1,...,k} |\psi(\lambda_i)|.$$

Components in the direction of unwanted eigenvectors are dumped.

The acceleration techniques and hybrid methods presented by Saad [16] attempt to improve the explicitly restarted Arnoldi iteration by approximately solving this min-max problem. Motivated by Manteuffel's scheme [12], Saad proposed the use of Chebyshev polynomials. A Chebyshev polynomial $\psi(A)$ on an ellipse containing the unwanted Ritz

values is applied to the restart vector in an attempt to accelerate convergence of the original explicitly restarted Arnoldi iteration. The polynomial is applied with the use of the familiar three-term recurrence.

3.1. The least-squares based method. The choice of ellipses as enclosing regions in Chebyshev acceleration may be overly restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance with an ellipse. This has spurred much research in which the acceleration polynomial is chosen so as to minimize an L_2 norm of the polynomial ψ on the boundary of the convex hull of the unwanted eigenvalues with respect to some suitable weight function ω. The only restriction with this technique is that the degree of the polynomial is limited because of cost and storage requirements. This, however, is overcome by compounding low degree polynomials. The stability of the computation is enhanced by employing a Chebyshev basis. It has been shown that the least-squares based method for solving linear systems is competitive with the ellipse based methods and are more reliable [17][13].

For convenience we can always normalize the polynomial so that $\psi(\lambda_1) = 1$. The desired polynomial satisfying the above constraint can be sought in the form $\psi_n(\lambda) \equiv 1 - \lambda s_n(\lambda)$. By the maximum principle, the maximum modulus of $|1-\lambda s_n(\lambda)|$ is found on the boundary of some region H of the complex plane that includes the spectrum of A and it is sufficient to regard the problem as being defined on the boundary. We use the least squares residual polynomial minimizing the L_2 norm $\parallel 1 - \lambda s_n(\lambda) \parallel_w$ with respect to some weight $w(\lambda)$ on the boundary of H [17]. Suppose that the $\mu + 1$ points $h_0, h_1, \cdots, h_\mu$ constitute the vertices of H. On each edge E_ν, $\nu = 1, \cdots, \mu$, of the convex hull, we choose a weight function $w_\nu(\lambda)$. Denoting by c_ν the center of the νth edge and by d_ν the half width, i.e., $c_\nu = (h_\nu + h_{\nu-1})/2$, $d_\nu = (h_\nu - h_{\nu-1})/2$, the weight function on each edge is defined by $w_\nu(\lambda) = 2|d_\nu^2 - (\lambda - c_\nu)^2|^{-\frac{1}{2}}/\pi$. The inner product on the space of complex polynomials is defined by $\langle p, q\rangle = \sum_{\nu=1}^{\mu} \int_{E_\nu} p(\lambda)\overline{q(\lambda)}w_\nu(\lambda)|d\lambda|$. An algorithm using explicitly the modified moments $\langle t_i(\lambda), t_j(\lambda)\rangle$, where $\{t_j\}$ is some suitable basis of polynomials, is developed for the problem of computing the least squares polynomials in the complex plane.

We express the polynomial $t_j(\lambda)$ in terms of the Chebyshev polynomials $t_j(\lambda)$ $= \sum_{i=0}^{j} \gamma_{i,j}^{(\nu)} T_i(\xi)$ where $\xi = (\lambda - c_\nu)/d_\nu$ is real. The expansion coefficients $\gamma_{i,j}^{(\nu)}$ can be computed easily from the three term recurrence of the polynomials $\beta_{k+1}t_{k+1}(\lambda) = (\lambda - \alpha_k)t_k(\lambda) - \delta_k t_{k-1}(\lambda)$. The problem $\min_{s\in\psi_{n-1}} \parallel 1-\lambda s_n(\lambda) \parallel_w$ is to find $\eta = (\eta_0, \eta_1, \cdots, \eta_{n-1})^T$ of $s_n(\lambda) = \sum_{i=0}^{n-1} \eta_i t_i(\lambda)$ so that $J(\eta) = \parallel 1 - \lambda s_n(\lambda) \parallel_w$ is minimum.

3.2. Approach. In the previous section we described the outline of the least-squares based method on any arbitrary area. It has a difficulty on the application to other purposes due to the constraint $\psi_n(0) = 1$.

We use the fact that the eigenvalue problem does not require any such condition to the polynomial and propose a new simple algorithm to get the mini-max polynomial to accelerate the convergence of the projection method. The minimum property of the Chebyshev functions described below is important to prove the optimality of this polynomial.

Let a non-negative weight function $w(\lambda)$ be given in the interval $a \geq \lambda \geq b$. The orthogonal polynomials $p_0(\lambda), p_1(\lambda), \cdots$, when multiplied by suitable factors C, possess a minimum property:

the integral $\int(\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_0)^2 w(\lambda)d\lambda$ takes on its least value when the polynomial in the integrand is $Cp_n(\lambda)$. The polynomial in the integrand may be written as a linear combination of the $p_i(\lambda)$, in the form $(Cp_n(\lambda) + c_{n-1}p_{n-1}(\lambda) + \cdots c_0)$. Since the functions $p_n(\lambda)\sqrt{w(\lambda)}$ are orthogonal, and in fact, orthogonal if the $p_i(\lambda)$ are appropriately defined, the integral is equal to $C^2 + \sum_{\nu=0}^{n-1} c_\nu^2$, which assumes its minimum at $c_0 = c_1 = \cdots = c_{n-1} = 0$.

Using the above property, we describe the new method to generate the coefficients of the ortho-normal polynomials in terms of the Chebyshev weight below.

We use the three term recurrence $\beta_{n+1}p_{n+1}(\lambda) = (\lambda - \alpha_n)p_n(\lambda) - \beta_n p_{n-1}(\lambda)$, where $p_i(\lambda)$ satisfies the ortho-normality. Because of the condition of the use of the Chebyshev polynomial $\psi_n(\lambda) = \sum_{i=0}^{n} \gamma_{i,n}^{(\nu)} T_i[(\lambda - c_\nu)/d_\nu]$, the constraints $\langle\psi_0, \psi_0\rangle = 2\sum_{\nu=1}^{\mu} |\gamma_{0,0}^{(\nu)}|^2 = 1$, $\langle\psi_1, \psi_1\rangle = \sum_{\nu=1}^{\mu}[2|\gamma_{0,1}^{(\nu)}|^2 + |\gamma_{1,1}^{(\nu)}|^2] = 1$, and $\langle\psi_0, \psi_1\rangle = 2\sum_{\nu=1}^{\mu} \gamma_{0,0}^{(\nu)}\overline{\gamma}_{1,1}^{(\nu)} = 0$ must hold. Moreover each expansion of $\psi_i(\lambda)$ at each edge must be consistent.

Using the three term recurrence of the Chebyshev polynomials, a similar recurrence $\beta_{k+1}\psi_{k+1}(\lambda) = (\lambda - \alpha_k)\psi_k(\lambda) - \delta_k\psi_{k-1}(\lambda)$ on $\psi_i(\lambda)$ holds. Denoting ξ_ν by $\xi_\nu = (\lambda - c_\nu)/d_\nu$, the equation can be rewritten as

$$\beta_{k+1}\psi_{k+1}(\lambda) = (d_\nu\xi + c_\nu - \alpha_k)\sum_{i=0}^{k} \gamma_{i,k}^{(\nu)} T_i(\xi) - \delta_k \sum_{i=0}^{k-1} \gamma_{i,k-1}^{(\nu)} T_i(\xi).$$

From the relations $\xi T_i(\xi) = [T_{i+1}(\xi) + T_{i-1}(\xi)]/2,\ i > 0$ and $\xi T_0(\xi) = T_1(\xi)$, it is expressed by

$$\sum \gamma_i \xi T_i(\xi) = \frac{1}{2}\gamma_1 T_0(\xi) + (\gamma_0 + \frac{1}{2}\gamma_2)T_1(\xi) + \cdots + \frac{1}{2}(\gamma_{i-1} + \gamma_{i+1})T_i(\xi) + \cdots + \frac{1}{2}(\gamma_{n-1} + \gamma_{n+1})T_n(\xi),$$

where $\gamma_{n+1} = 0$, and arranged into

$$\beta_{n+1}\psi_{n+1}(\lambda) = d_\nu[\frac{\gamma_{1,n}^{(\nu)}}{2}T_0(\xi) + (\gamma_{0,n}^{(\nu)} + \frac{\gamma_{2,n}^{(\nu)}}{2})T_1(\xi) + \cdots + \sum_{i=2}^{n} \frac{1}{2}(\gamma_{i-1,n}^{(\nu)} + \gamma_{i+1,n}^{(\nu)})T_i(\xi)]$$

$$+(c_\nu - \alpha_n)\sum_{i=0}^{n} \gamma_{i,n}^{(\nu)} T_i(\xi) - \delta_n \sum_{i=0}^{n-1} \gamma_{i,n-1}^{(\nu)} T_i(\xi) \qquad (T_{-1} = T_1).$$

Comparing the equation with $\psi_{n+1}(\lambda) = \sum_{i=0}^{n+1} \gamma_{i,n+1}^{(\nu)} T_i(\xi)$, we find the following relations

$$\beta_{n+1}\gamma_{0,n+1}^{(\nu)} = \frac{1}{2}d_\nu\gamma_{1,n}^{(\nu)} + (c_\nu - \alpha_n)\gamma_{0,n}^{(\nu)} - \delta_n\gamma_{0,n-1}^{(\nu)},$$

$$\beta_{n+1}\gamma_{1,n+1}^{(\nu)} = d_\nu(\gamma_{0,n}^{(\nu)} + \frac{1}{2}\gamma_{2,n}^{(\nu)}) + (c_\nu - \alpha_n)\gamma_{1,n}^{(\nu)} - \delta_n\gamma_{1,n-1}^{(\nu)},$$

and

$$\beta_{n+1}\gamma_{i,n+1}^{(\nu)} = \frac{d_\nu}{2}[\gamma_{i+1,n}^{(\nu)} + \gamma_{i-1,n}^{(\nu)}] + (c_\nu - \alpha_n)\gamma_{i,n}^{(\nu)} - \delta_n\gamma_{i,n-1}^{(\nu)}$$

$$i = 2, ..., n+1 \qquad (\gamma^{(\nu)}_{-1,n} = \gamma^{(\nu)}_{1,n}, \quad \gamma^{(\nu)}_{i,n} = 0 \quad i > n).$$

Using the relation $\beta_{k+1}\psi_{k+1}(\lambda) = (\lambda - \alpha_k)\psi_k(\lambda) - \delta_k\psi_{k-1}(\lambda)$ and the orthogonality of the Chebyshev polynomials, we derive

$$\beta_{k+1} = \langle \psi_{k+1}, \psi_{k+1} \rangle^{1/2} = \sum_{\nu=1}^{\mu} \int_{E_\nu} \psi_{k+1}\overline{\psi_{k+1}} w_\nu(\lambda)|d\lambda| = \sum_{\nu=1}^{\mu} {\sum_{i=0}^{k+1}}' \gamma^{(\nu)}_{i,k+1}\overline{\gamma^{(\nu)}_{i,k+1}}$$

where we denote by $\sum'^{n}_{i=0} a_i = 2a_0 + \sum_{i=1}^{n} a_i$.

α and δ are computed similarly:

$$\alpha_k = \langle \lambda\psi_k, \psi_k \rangle = \sum_{\nu=1}^{\mu} (c_\nu {\sum_{i=0}^{k}}' \gamma^{(\nu)}_{i,k}\overline{\gamma^{(\nu)}_{i,k}} + d_\nu {\sum_{i=0}^{k}}' \gamma^{(\nu)}_{i,k}\overline{\gamma^{(\nu)}_{i+1,k}}), \quad \delta_k = \langle \lambda\psi_k, \psi_{k-1} \rangle = \sum_{\nu=1}^{\mu} d_\nu \upsilon_\nu$$

where $\upsilon_\nu = \gamma^{(\nu)}_{1,k}\overline{\gamma^{(\nu)}_{0,k-1}} + (\gamma^{(\nu)}_{0,k} + \frac{1}{2}\gamma^{(\nu)}_{2,k})\overline{\gamma^{(\nu)}_{1,k-1}} + \sum_{i=2}^{k-1} \frac{1}{2}(\gamma^{(\nu)}_{i-1,k} + \gamma^{(\nu)}_{i+1,k})\overline{\gamma^{(\nu)}_{i,k-1}}$.

4. Evaluation.

4.1. Complexity of the algorithms. The cost in terms of the number of floating-point operations are as follows: We denote by n, nz, m, r, k respectively the order of the matrix, its number of nonzero entries, the number of block Arnoldi steps, the number of required eigenvalues, and the degree of the Chebyshev polynomial. The block Arnoldi method costs $\sum_{j=1}^{m}\{2r\,nz + 4nr^2 j + 2r(r+1)n\} = 2rm\,nz + 2mr(mr + 2r + 1)n$ flops. $10r^3m^3$ flops are required for the computation of the eigenvalues of H_m of order mr by the QR method, $r^3\mathcal{O}(m^2)$ for the corresponding eigenvectors by the inverse iteration, and $2kr\,nz + \mathcal{O}(n)$ for the Chebyshev iteration [7, 18]. The computation of the coefficients costs approximately $\mathcal{O}(\mu k^2)$ flops, where μ is the number of the vertices of the convex hull.

4.2. Numerical results. This section reports the results of the numerical experiments of our method and evaluates its performance. The experiments are performed on HP9000/720 using double precision.

We start with the decision of each element of the matrix given in the problem. In this section, the scaled sequences of random numbers are assigned respectively to the real and the imaginary parts of the eigenvalues except for those which are to be selected. The matrices are block diagonals with 2×2 or 1×1 diagonal blocks. Each block is of the form $\begin{bmatrix} a & b/2 \\ -2b & a \end{bmatrix}$ to prevent the matrix to be normal and has eigenvalues $a \pm bi$. It is transformed by an orthogonal matrix generated from a matrix with random elements by the Schmidt's orthogonalization method. m and n_c denote the order of the Arnoldi method and the maximum order of the Chebyshev polynomials respectively. We compare this algorithm with the double-shifted QR method. The error is computed by the L_2 norm.

In this section we test the some variations of the distribution of the eigenvalues using the matrices of order 50, the cases of $\lambda_{\max} = 2, 1.5$, and 1.1 while the distribution of the other eigenvalues is $\Re\mathrm{e}\,\lambda \in [0, 1]$, and $\Im\mathrm{m}\,\lambda \in [-1, 1]$. We denote the number of the iterations by n_{iter}.

4.3. Comparison with other methods. Some test problems from the Harwell-Boeing sparse matrix collection [4], the spectral portraits [6] of which are shown in Figure 1 and Figure 2, are solved using the block Arnoldi method. Ho's algorithm is used for reference.

The stopping criterion is based on the maximum of all computed residuals $\max_{1\le i\le r} \| Ax_i - \lambda_i x_i \|_2 / \| x_i \|_2 \equiv \max_{1\le i\le r} \| H_{m+1,m} Y_{m,r,i} \|_2 / \| Y_{m,i} \|_2 \le \epsilon$. $Y_{m,r,i}$ and $Y_{m,i}$ stand for the i-th column of the $Y_{m,r}$ and Y_m.

Table 4 and Table 5 indicate that Ho's algorithm shows better performance than the orthogonality-based method in most conditions except for the cases where the moduli of the necessary eigenvalues are much larger than those of the unnecessary eigenvalues. We may derive from the result the poor optimality of the convex hull despite its low computation cost.

Lehoucq and Scott [10] presented a software survey of large-scale eigenvalue methods and comparative results. The Arnoldi-based software included the following three packages `ARNCHEB` package [3], the `ARPACK` software package [11], and the Harwell Subroutine Library code `EB13` [19].

The `ARNCHEB` package provides the subroutine `ARNOL`, which implements an explicitly restarted Arnoldi iteration. The code is based on the deflated polynomial accelerated Arnoldi iteration and uses Chebyshev polynomial acceleration. The Harwell Subroutine Library code `EB13` implements the similar algorithm and also uses Ho's Chebyshev polynomial acceleration. The `ARPACK` provides subroutine `DNAUPD` that implements the implicitly restarted Arnoldi iteration.

Some findings are reported on these methods:

- `ARNCHEB` gives reasonable results for computing a single eigenpair but it can struggle on problems for which several eigenvalues are requested.
- `ARPACK` displays monotonic consistency and is generally the fastest and most dependable of the codes studied, especially for small convergence tolerances and large departures from normality. It uses dramatically fewer matrix-vector product than `ARNCHEB`. However, its restarting strategy can be more expensive.

Moreover, from the results of Table 6 and Table 7, we can derive the strong dependency of `EB13` on the distribution of spectrum.

TABLE 3

The distribution of the other eigenvalues: $\Re\mathrm{e}\,\lambda \in [0,1]$, $\Im\mathrm{m}\,\lambda \in [-1,1]$. *CPU times (in seconds) by HP9000/720.*

maximum	orthogonality-based					Arnoldi				QR	
eigenvalues	n_{iter}	m	n_c	error	time	n_{iter}	m	error	time	error	time
2	2	5	15	3.6E-15	0.38	2	15	8.9E-16	0.57	5.1E-15	1.87
1.5	3	5	20	3.0E-15	0.70	3	15	3.7E-15	0.82	3.6E-15	1.85
1.1	5	10	20	2.9E-14	1.6	1	50	7.5E-13	3.93	5.2E-15	18.8

5. Parallelization. The parallelization of non-Hermitian eigenproblem is not commonly studied. A MIMD parallel implementation of the Arnoldi method is implemented and mentioned in [14] for both tightly coupled as well as loosely coupled memory machines with vector elementary processors and large granularity. This study has already shown

TABLE 4

*Test problems from CHEMWEST, a library in the Harwell-Boeing Sparse Matrix Collection, which was extracted from modeling of chemical engineering plants. The results by Ho's algorithm (right) versus those by the orthogonality-based method (left), with size of the basis 20, degree of the polynomial 20, and block size 1, respectively, are listed. * denotes the algorithm fails to converge.*

problem	WEST0497		WEST0655		WEST0989		WEST2021	
order of matrix	497		655		989		2021	
number of entries	1727		2854		3537		7353	
number of multiplications	924	440	275	120	13751	*	767	320
number of restarts	14	10	3	2	162	*	12	7
CPU time (sec.)	0.37	0.22	0.17	0.12	8.71	*	1.28	0.67

TABLE 5

Test problems from TOLOSA extracted from fluid-structure coupling (flutter problem). Size of the basis, degree of the polynomial, and block size are 20, 20, 1, respectively.

order of matrix	2000		4000		6000		8000		10000	
number of entries	5184		8784		12384		15984		19584	
number of multiplications	589	240	393	180	236	140	393	380	236	80
number of restarts	7	4	5	3	3	2	5	7	3	1
CPU time (sec.)	0.83	0.43	1.24	0.70	1.23	0.85	2.57	2.81	2.14	0.97

TABLE 6

Evaluation by Lehoucq and Scott. CPU times (in seconds) by IBM RS/6000 3BT and matrix-vector products for computing the right-most eigenvalues of WEST2021 from CHEMWEST (denotes convergence not reached within* $2000m$ *matrix-vector products). We denote by* r *the block size and by* m *the subspace dimension.*

Algorithm	r=1,m=8	r=5,m=20
`EB12`	*	98/20930
`ARNCHEB`	8.6/3233	71/15921
`EB13`	17/4860	18/4149
`ARPACK`	3.7/401	2.1/167

TABLE 7

CPU times (in seconds) and matrix-vector products for computing the right-most eigenvalues of PORES2, matrix of order 1224 with 9613 entries, which was extracted from reservoir simulation.

Algorithm	r=1,m=12	r=4,m=20
`EB12`	0.6/423	9.1/2890
`ARNCHEB`	3.4/1401	4.7/1712
`EB13`	0.4/119	1.3/305
`ARPACK`	0.5/90	1.3/151

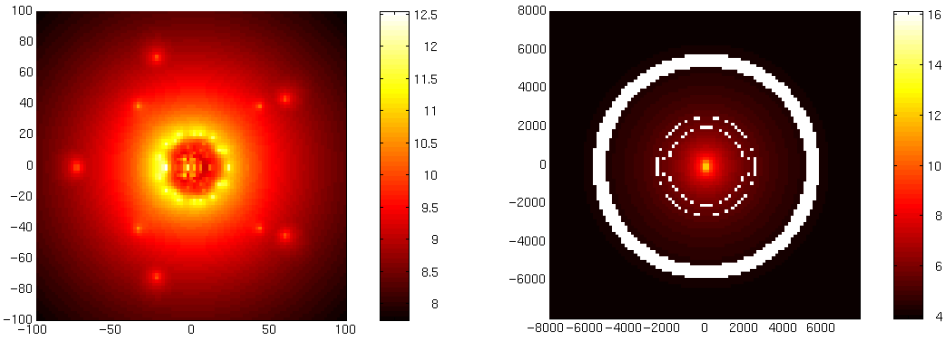

FIG. 1. *Spectral Portraits of WEST0655 and WEST0989 from CHEMWEST.*

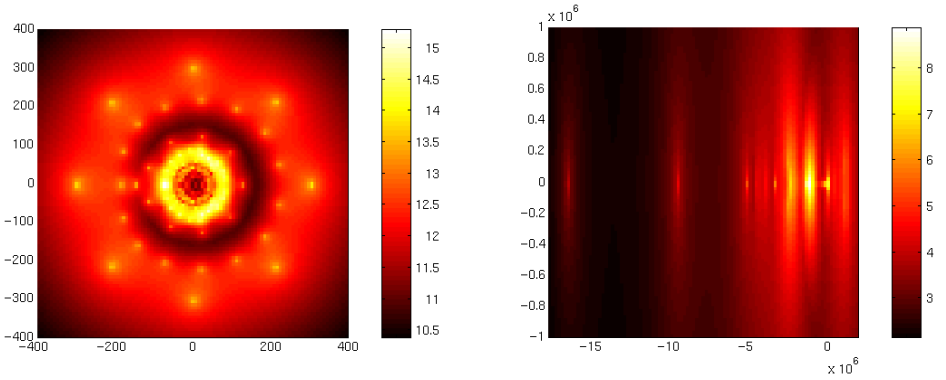

FIG. 2. *Spectral Portraits of WEST2021 and PORES2*

that the QR method is the most significant bottleneck on these MIMD architectures. The speed of convergence for such methods usually increases which the subspace size m is chosen larger. The number of floating-point operations, and therefore the time required by the algorithm, rapidly increases with subspace dimension m. Furthermore, m must be taken as small as possible to avoid QR to become a bottleneck.

Henry and van de Geijn [8] show that under certain conditions the described approach is asymptotically 100% efficient. It is impossible to find an implementation with better scalability properties, since for maintaining a given level of efficiency the dimension of the matrix must grow linearly with the number of processors. Therefore, it will be impossible to maintain the performance as processors are added, since memory requirements grow with the square of the dimension, and physical memory grows only with the number of processors. They also show that for the standard implementation of the sequential QR algorithm, it is impossible to find an implementation with better scalability properties.

6. Conclusion. We simplified the computation of the least-squares polynomial which minimizes its norm on the boundary of the convex hull enclosing unwanted eigenvalues, using the minimum property of the orthogonal polynomials. This method requires the computation of $2rm\,nz + 2mr(mr + 2r + 1)n$ flops for the block Arnoldi method, $r^3[10m^3 + \mathcal{O}(m^2)]$ for the computation of the eigenvalues of H_m, and $2kr\,nz + \mathcal{O}(n)$ for the Chebyshev iteration. The number of floating point operations rapidly increases with the size of the subspace dimension m and it indicates that we need to take m as small as possible if we want to avoid QR to become a bottleneck, even on parallel architectures. Although some problems are to be solved, the validity of our method was confirmed by the experiments using the Harwell-Boeing Sparse Matrix Collection, which is a set of standard test matrices for sparse matrix problems. A more detailed analysis of the precision and the complexity of the methods is required.

REFERENCES

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

[2] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, Tech. Report 95-04, Department of Mathematics, University of Kentucky, 1995.

[3] T. Braconnier, *The Arnoldi-Tchebycheff algorithm for solving nonsymmetric eigenproblems*, Tech. Report TR/PA/93/25, CERFACS, Toulouse, 1993.

[4] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Softw., 15 (1989), pp. 1–14.

[5] D. R. Fokkema, G. L. G. Sleijpen, and H. A. V. der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, Tech. Report 941, Department of Mathematics, Utrecht University, 1996.

[6] V. Fraysse, L. Giraud, and V. Toumazou, *Parallel computation of spectral portrait on the Meiko CS2*, in High-Performance Computing and Networking, H. Liddell, A. Colbrook, B. Hertzberger, and P. Sloot, eds., vol. 1067, Springer-Verlag, pp. 312–318.

[7] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.

[8] G. Henry and R. van de Geijn, *Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problems: myth and reality*, vol. 17, 1996.

[9] R. B. Lehoucq, *Restarting an Arnoldi Reduction*, Tech. Report MCS-P591-0496, Argonne National Laboratory.
[10] R. B. Lehoucq and J. A. Scott, *An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices*, Tech. Report MCS-P547-1195, Argonne National Laboratory.
[11] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM Journal on Matrix Analysis and Applications, 17 (1996).
[12] T. A. Manteuffel, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.
[13] A. Nishida and Y. Oyanagi, *A Polynomial Acceleration of the Projection Method for Large Nonsymmetric Eigenvalue Problems*, in Kokyuroku, vol. 944, Research Institute for Mathematical Sciences, Kyoto University, 1996, pp. 135–146.
[14] S. G. Petiton, *Parallel subspace method for non-Hermitian eigenproblems on the Connection Machine (CM2)*, Applied Numerical Mathematics, 10 (1992), pp. 19–35.
[15] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra and its Applications, 34 (1980), pp. 269–295.
[16] ——, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math.Comp., 42 (1984), pp. 567–588.
[17] ——, *Least squares polynomials in the complex plane and their use for solving nonsymmetric linear systems*, SIAM J. Numer. Anal., 24 (1987), pp. 155–169.
[18] M. Sadkane, *A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.
[19] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices*, ACM Trans. Mathematical Software, (1995), pp. 432–475.
[20] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.