LEAST SQUARES ARNOLDI FOR LARGE NONSYMMETRIC EIGENPROBLEMS

AKIRA NISHIDA*

Abstract. In this paper, we propose a highly efficient accelerating method for the restarted Arnoldi iteration to compute the eigenvalues of a large nonsymmetric matrix. Its effectiveness is proved by various numerical experiments and comparisons with other approaches. Several new results on the characteristics of the polynomial acceleration are also reported.

The Arnoldi iteration has been the most popular method for nonsymmetric large eigenproblems. Although the defect of increasing computational complexity per iteration step can be improved with the explicitly restarting technique, by which the dimensions of the Krylov subspaces are kept modest, the dimension of the subspace becomes large, in particular when the required eigenvalues are clustered. In our study, an accelerating polynomial is chosen to minimize an L_2 norm of the polynomial on the boundary of the convex hull with respect to some suitable weight function. A new simple algorithm is proposed for the efficient computation of the mini-max polynomial to accelerate the convergence of the Arnoldi iteration.

From the numerical results, we can derive the strong dependency of the polynomial acceleration on the distribution of spectrum, which proves the better performance of our algorithm than the Chebyshev acceleration, in the cases where the moduli of the wanted eigenvalues are considerably larger than those of the unwanted eigenvalues, and the faster convergence than those of all the other approaches, especially when the shape of the convex hull of the unwanted eigenvalues bears little resemblance with an ellipse.

Finally, we propose a new parallelization technique for the nonsymmetric double shifted QR algorithm with perfect load balance and uninterrupted pipelining on distributed memory parallel architectures, which is strongly required from the viewpoint of complexity of the Arnoldi iteration. Its parallel efficiency is much higher than those reported in other papers.

1. Introduction. In the last few years, there have been great progress in the developments of the methods for the standard eigenproblem. Arnoldi's method, which have the disadvantage of increasing computational complexity per iteration step, was improved with the restarting technique, by which the dimensions of the Krylov subspaces is kept modest.

Although the Arnoldi iteration is a considerably effective solution, the dimension of the subspace becomes excessively large, especially when the required eigenvalues are clustered. Furthermore, outer eigenvalues on the envelope of the spectrum show faster convergence. This difficulty has been overcome by using the polynomial acceleration technique, which is an extension of the similar technique for symmetric matrices. In the nonsymmetric case, we consider the distribution of the eigenvalues in the complex plane. Suppose $A \in \mathbf{R}^{n\times n}$ is a diagonalizable matrix with eigensolutions (u_j, λ_j) for $j = 1, ..., n$. Letting $p(\cdot)$ be some polynomial, the current starting vector x_1 can be expanded as $p(A)x_1 = c_1 p(\lambda_1)u_1 + \cdots + c_n p(\lambda_n)u_n$ in terms of the basis of eigenvectors. Then if we assume that the eigenvalues are ordered so that the wanted k ones are located at the beginning of the expansion, we seek a polynomial such that $\max_{i=k+1,...,n} |p(\lambda_i)| < \min_{i=1,...,k} |p(\lambda_i)|$ holds.

The acceleration techniques attempt to improve the restarted Arnoldi iteration by solving this min-max problem, where a Chebyshev polynomial $p(A)$ on an ellipse containing the unwanted Ritz values is applied to the restart vector to accelerate convergence of the restarted Arnoldi iteration.

2. Arnoldi's Method.

2.1. Algorithm. Suppose $A \in \mathbf{R}^{n\times n}$. The Arnoldi approach involves the column-by-column generation of an orthogonal matrix Q such that $Q^T AQ = H$ is the Hessenberg reduction. If we write Q as $[q_1, ..., q_m] \in \mathbf{R}^{n\times m}$ and isolate the last term in the summation $Aq_m = \sum_{i=1}^{m+1} h_{im} q_m$, then we have

$$h_{m+1,m} q_{m+1} = Aq_m - \sum_{i=1}^{m} h_{im} q_m \equiv r_m \tag{1}$$

where $h_{im} = q_i^T Aq_m$ for $i = 1, ..., m$. We assume that q_1 is a given 2-norm starting vector.

PROPOSITION 2.1. *The Arnoldi process computes an orthonormal basis for the Krylov subspace* $\mathsf{K}_m(A, q_1)$

$$\mathrm{span}\{q_1, ..., q_m\} = \mathrm{span}\{q_1, Aq_1, ..., A^{m-1}q_1\}, \tag{2}$$

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

* Department of Information Science, Faculty of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113 Japan.

Proof. The vectors q_j for $j = 1, 2, ..., m$ are orthonormal by construction. That they span K_m follows from the fact that each vector v_j is of the form $p_{j-1}(A)v_1$ where p_{j-1} is a polynomial of degree $j-1$, which can be shown by induction on j. □

ALGORITHM 2.1 (ARNOLDI).

1. $h_{1,1} = (Aq_1, q_1)$
2. for $j = 1, ..., m-1$, put
3. $\quad r_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i, \quad h_{j+1,j} = \| r_j \|_2$
4. $\quad q_{j+1} = h_{j+1}^{-1} r_j, \quad h_{i,j+1} = (Aq_{j+1}, q_i), \quad (i \leq j+1).$

PROPOSITION 2.2. *Denote by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations*

$$AQ_m = Q_m H_m + r_m e_m^T \tag{3}$$

$$Q_m^H A Q_m = H_m \tag{4}$$

hold, where $e_m = (0, ..., 0, 1)^T$.

Proof. (3) holds from the equality $Aq_j = \sum_{i=1}^{j+1} h_{ij} q_i, j = 1, 2, ..., m$. (4) follows by multiplying both sides of (3). □

The algorithm terminates when $r_j = 0$, which is impossible if the minimal polynomial of A with respect to q_1 is of degree $\geq m$. If this condition is satisfied, H_m is an irreducible Hessenberg matrix.

A complete reduction of A to Hessenberg form can be written as $A = QHQ^H$. Consider the first $m < n$ columns of $AQ = QH$. Let Q_m be the $n \times m$ matrix whose columns are the first m columns of Q, and let $\tilde{H}_m$ be the $(m+1) \times m$ upper-left section of H_{m+1}. We have $AQ_m = Q_{m+1}\tilde{H}_m$ and the mth column of this equation can be written as

$$Aq_m = h_{1m} q_1 + \cdots + h_{mm} q_m + h_{m+1,m} q_{m+1}. \tag{5}$$

Then the vectors $\{q_i\}$ form bases of the successive Krylov subspaces generated by A and b, defined as

$$\mathsf{K}_m = \text{span}\{b, Ab, ..., A^{m-1}b\} = \text{span}\{q_1, q_2, ..., q_m\} \subseteq \mathbf{C}^n. \tag{6}$$

Since the vectors q_i are orthonormal, these are orthonormal bases. Let us define K_m to be a $n \times m$ Krylov matrix

$$K_m = [b, Ab, \cdots, A^{m-1}b]. \tag{7}$$

Then K_m have a reduced QR factorization $K_m = Q_m R_m$, where Q_m is the same matrix as above, and might be expected to contain good information about the eigenvalues of A with largest modulus.

PROPOSITION 2.3. *Let $y_i \in \mathbf{C}^m$ be an eigenvector of H_m associated with the eigenvalue $\tilde{\lambda}_i$ and $\tilde{x}_i = Q_m y_i$. Then*

$$(A - \tilde{\lambda}_i I)\tilde{x}_i = h_{m+1,m} e_m^H y_i q_{m+1} \tag{8}$$

and, therefore,

$$\| (A - \tilde{\lambda}_i I)\tilde{x}_i \|_2 = h_{m+1,m} |e_m^H y_i|. \tag{9}$$

Proof. (8) follows from multiplying both sides of (3) by y_i. □

2.2. Block Arnoldi. Suppose that we are interested in computing the r eigenvalues of a matrix $A \in \mathbf{R}^{n \times n}$. Assume that $V_1 \in \mathbf{R}^{n \times r}$ is a rectangular matrix having r orthonormal columns. Then the algorithm of the block-Arnoldi method can be described as follows:

ALGORITHM 2.2 (BLOCK ARNOLDI).

1. For $k = 1, ..., m-1$, do
2. $\quad W_k = AV_k$
3. $\quad$ For $i = 1, ..., k$, do
4. $\quad\quad H_{i,k} = V_i^T W_k; \quad W_k = W_k - V_i H_{i,k}$
5. $\quad Q_k R_k = W_k$
6. $\quad V_{k+1} = Q_k; \;\; H_{k+1,k} = R_k$

Letting $U_m = [V_1, ..., V_m]$, the restriction of the matrix A to the Krylov subspace is written as

$$H_m = U_m^T A U_m = \begin{pmatrix} H_{1,1} & H_{1,2} & & \cdots & H_{1,m} \\ H_{2,1} & H_{2,2} & & & H_{2,m} \\ O & \ddots & \ddots & & \vdots \\ \vdots & \ddots & & & \\ O & \cdots & O & H_{m,m-1} & H_{m,m} \end{pmatrix}. \tag{10}$$

The above algorithm gives

$$AV_k = \sum_{i=1}^{k} V_i H_{i,k} + V_{k+1} H_{k+1,k}, \quad \text{for} k = 1, ..., m, \tag{11}$$

which can be written in a form as

$$AU_m = U_m H_m + [0, ..., 0, V_{m+1} H_{m+1,m}]. \tag{12}$$

Letting $\tilde{\Lambda}_m = \text{diag}(\tilde{\lambda}_1, ..., \tilde{\lambda}_{mr})$ be the diagonal matrix of eigenvalues of H_m corresponding to the eigenvectors $Y_m = [y_1, ..., y_{mr}]$, the above relation gives

$$AU_m Y_m - U_m H_m Y_m = [0, ..., 0, V_{m+1} H_{m+1,m}] Y_m. \tag{13}$$

If we denote by $\tilde{X}_m = U_m Y_m$ the matrix of approximate eigenvectors of A and by $Y_{m,r}$ the last r block of Y_m, we have

$$\| A\tilde{X}_m - \tilde{X}_m \tilde{\Lambda}_m \|_2 = \| H_{m+1,m} Y_{m,r} \|_2, \tag{14}$$

which will be used for the stopping criterion in the following numerical evaluation.

2.3. The Arnoldi Iteration. The Arnoldi process, which have the disadvantage of increasing computational complexity per iteration step, can be improved with the restarting technique, by which the dimensions of the Krylov subspaces is kept modest [4]. In the iterative variant, we start with an initial vectors V_1 and fix a moderate value m, then compute the eigenvectors of H_m. We begin again, using new starting vectors computed from the approximate eigenvectors.

The algorithm of the explicitly restarted Arnoldi iteration is summarized below. 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.

ALGORITHM 2.3 (EXPLICITLY RESTARTED ARNOLDI).

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

3. Least Squares Based Polynomial Acceleration.

3.1. Basic Approach. Let $\mathcal{T}$ be a simply connected region in the complex ζ-plane.

THEOREM 3.1 (THE MAXIMUM PRINCIPLE). *If a function $f(\zeta)$ is defined and continuous on a closed bounded set $\mathcal{T}$ and analytic on the interior of $\mathcal{T}$, then the maximum of $|f(\zeta)|$ on $\mathcal{T}$ is assumed on the boundary of $\mathcal{T}$.*

Using the above property, we can regard the mini-max problem as being defined on the boundary of the region which contains the spectrum. Let the boundary C of the region be a continuum consisting of a finite number of rectifiable Jordan arcs. The integrals considered will have the form

$$\int_C f(\zeta)|d\zeta|, \tag{15}$$

where $f(\zeta)$ is a Lebesque-integrable function defined on C and $|d\zeta|$ is the arc element on C.

DEFINITION 3.1. *The scalar product of two functions $f(\zeta)$ and $g(\zeta)$, ζ on C, is defined by the integral*

$$\langle f, g\rangle = \int_C f(\zeta)\bar{g}(\zeta)w(\zeta)|d\zeta|. \tag{16}$$

We introduce here the least squares residual polynomial minimizing an L_2 norm with respect to some weight $w(\zeta)$ on the boundary of a convex hull formed from the approximate eigenestimates. Note that the constraint $p_n(\zeta) \in \pi_n^1$ is not necessary for the eigenproblems [4].

The contour considered here is the finite union of line segments:

DEFINITION 3.2. *Denote by $\mathcal{H}$ the convex hull constituted from the μ vertices $h_0, \cdots, h_\mu$, and by*

$$\vartheta_\nu = \frac{1}{2}(h_\nu + h_{\nu-1}), \qquad \delta_\nu = \frac{1}{2}(h_\nu - h_{\nu-1}), \tag{17}$$

the center and the half width on each edge C_ν, $\nu = 1, 2, ..., \mu$, respectively. We can define the Chebyshev weight

$$w_\nu(\zeta) = \frac{2}{\pi}\left[\delta_\nu^2 - (\zeta - \vartheta_\nu)^2\right]^{-\frac{1}{2}}, \quad \zeta \in C_\nu \tag{18}$$

on each edge C_ν, and the inner product

$$\langle p, q\rangle = \int_C p(\zeta)\bar{q}(\zeta)w(\zeta)|d\zeta| \equiv \sum_{\nu=1}^{\mu}\int_{C_\nu} p(\zeta)\bar{q}(\zeta)w_\nu(\zeta)|d\zeta| = \sum_{\nu=1}^{\mu}\langle p, q\rangle_\nu \tag{19}$$

on the boundary C. A norm is defined by $\| p \|_w^2 = \langle p, p\rangle$.

Thus, we can rewrite the problem as

$$\min_{p\in\pi_n}\max_{\zeta\in\mathcal{H}}|p(\zeta)| = \min_{p\in\pi_n,\ \zeta\in C}\| p(\zeta) \|_w\ . \tag{20}$$

An algorithm using explicitly the modified moments $\langle t_i(\zeta), t_j(\zeta)\rangle$, where $\{t_j\}$ is some suitable basis of polynomials, is developed for the problem of computing the least squares polynomials in the complex plane. The set of Chebyshev polynomials suitably shifted and scaled is reasonable as the basis $\{t_j\}$ rather than the power basis $\{1, \zeta, \zeta^2, \cdots, \zeta^{n-1}\}$, which forces unstable computation. However, the matrix M_n whose elements $m_{i,j}$ are defined by

$$m_{i,j} = \langle t_{j-1}, t_{i-1}\rangle, \qquad i, j = 1, 2, \cdots n+1 \tag{21}$$

is still likely to become increasingly ill-conditioned as its size $n+1$ increases.

We express the polynomial $t_j(\zeta)$ in terms of the Chebyshev polynomials

$$t_j(\zeta) = \sum_{i=0}^{j}\varphi_{i,j}^{(\nu)}T_i(\xi_\nu), \quad \text{where } \xi_\nu = \frac{\zeta - \vartheta_\nu}{\delta_\nu} \text{ is real.} \tag{22}$$

The expansion coefficients $\varphi_{i,j}^{(\nu)}$ can be computed easily from the three term recurrence of the polynomials

$$\beta_{k+1}t_{k+1}(\zeta) = (\zeta - \alpha_k)t_k(\zeta) - \gamma_k t_{k-1}(\zeta). \tag{23}$$

3.2. Least Squares Arnoldi. In this section, we propose a new algorithm to get the mini-max polynomial for the accelerating the Arnoldi iteration.

We can orthogonalize the system and lead to a set of polynomials $\{p_n(\zeta)\}$, where

(a) $p_n(\zeta)$ is a polynomial of degree n in which the coefficients of ζ^n is real and positive;

(b) the system $\{p_n(\zeta)\}$ is orthonormal, that is,

$$\int_C p_n(\zeta)\bar{p}_m(\zeta)w(\zeta)|d\zeta| = \delta_{nm}, \qquad n, m, = 0, 1, 2, ... \tag{24}$$

DEFINITION 3.3. *Let $f(\zeta)$ be a continuous function defined on C and let there correspond the formal Fourier expansion*

$$f(\zeta) \sim f_0 p_0(\zeta) + f_1 p_1(\zeta) + \cdots + f_n p_n(\zeta) + \cdots. \tag{25}$$

The coefficients f_n, called the Fourier coefficients of $f(\zeta)$ with respect to the given system, are defined by

$$f_n = \langle f, p_n \rangle = \int_C f(\zeta)\bar{p}_n(\zeta)w(\zeta)|d\zeta|, \qquad n = 0, 1, 2, ... \tag{26}$$

THEOREM 3.2 (BESSEL'S INEQUALITY). *The partial sums $s_n(\zeta)$ of (25) minimize the integral*

$$\int_C |f(\zeta) - \rho(\zeta)|^2 w(\zeta)|d\zeta| \tag{27}$$

if $\rho(\zeta)$ ranges over the class of all π_n. The minimum is

$$\int_C |f(\zeta)|^2 w(\zeta)|d\zeta| - |f_0|^2 - |f_1|^2 - \cdots |f_n|^2. \tag{28}$$

This also yields Bessel's inequality

$$|f_0|^2 + |f_1|^2 + \cdots + |f_n|^2 \leq \| f(\zeta) \|_w = \int_C |f(\zeta)|^2 w(\zeta)|d\zeta|. \tag{29}$$

Proof. For any finite system of complex numbers $c_0, c_1, c_2, ..., c_n$, we have $||f - \sum_{i=1}^n c_i p_i||_w^2 = (f - \sum_{i=1}^n c_i p_i, f - \sum_{i=1}^n c_i p_i) = \| f \|_w^2 - \sum_{i=1}^n c_i \bar{f}_i - \sum_{i=1}^n \bar{c}_i f_i + \sum_{i=1}^n |c_i|^2 = \| f \|_w^2 - \sum_{i=1}^n |f_i|^2 + \sum_{i=1}^n |f_i - c_i|^2$ by orthonormaliby of $\{p_n\}$. Since its minimum is attained when $c_i = f_i$ $(i = 1, 2, ..., n)$, we have $\| f - \sum_{i=1}^n c_i p_i \|_w^2$, and hence $\sum_{i=1}^n |f_i|^2 \leq \| f \|_w^2$. □

The Theorem 3.2 has the following important consequence:

COROLLARY 3.1. *An orthonormal system $\{p_n(\zeta)\}$ satisfies the condition (20).*

Using the property, we can directly generate the coefficients of the ortho-normal polynomials in terms of the Chebyshev weight, on the basis of the three term recurrence, where each polynomial satisfies the condition (24) of orthonormality on $p(\zeta)$. Note that the expansion of $p_i(\zeta)$ at each edge must be consistent.

4. Evaluation. Denoting the number of nonzero entries in A by n_{nz} and the number of required eigenvalues in the block Arnoldi iteration by r, the cost of block Arnoldi method can be defined as $\mathcal{O}(rmn_{nz} + m^2r^2n)$ flops. $10r^3m^3$ flops are required for the computation of the eigenvalues of H_m of order mr by the QR algorithm, $r^3\mathcal{O}(m^2)$ for the corresponding eigenvectors by the inverse iteration, and $2kr\, n_{nz} + \mathcal{O}(n)$ for the Chebyshev iteration. 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. The complexity of the orthogonality-based method is roughly $\mathcal{O}(n^2)$, while that of the QR algorithm is $\mathcal{O}(n^3)$.

We solved some test problems from the Harwell-Boeing sparse matrix collection, using the block Arnoldi iteration. Table 1 and Figure 1 indicates that our algorithm shows better performance than the ellipse based method in the cases where the moduli of the wanted eigenvalues are considerably larger than those of the unwanted eigenvalues. Table 2 shows the comparative results on the ARPACK software package and the Harwell Subroutine

Library code EB13 [2]. EB13 and ARPACK implement the explicitly restarted Arnoldi iteration , the ellipse based Chebyshev polynomial acceleration, and the implicitly restarted Arnoldi iteration, respectively. From the results of Table 2, we can derive the strong dependency of the polynomial acceleration on the distribution of spectrum. Figure 1 and some additional results on the transition of accelerating polynomials [3] indicate that the non-clustered distribution of spectra causes slow convergence, which is due to the discrepancies between the accelerated domains and the computed spectra. ARPACK displays monotonic consistency and is generally faster and more dependable for small convergence tolerances and large departures from normality. However, its restarting strategy can be more expensive.

TABLE 1

*Test problems extracted from the modeling of chemical engineering plants. The results by ellipse based 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. CPU time by Alpha Station 600 5/333.*

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 2

CPU times of explicitly and implicitly restarted Arnoldi iterations by IBM RS/6000 3BT and matrix-vector products for computing the right-most eigenvalues of WEST2021 and PORES2 of order 1224. We denote by r the number of eigenvalues and by m the subspace dimension.

WEST2021	r=1,m=8	r=5,m=20
EB13	17/4860	18/4149
ARPACK	3.7/401	2.1/167

PORES2	r=1,m=12	r=4,m=20
EB13	0.4/119	1.3/305
ARPACK	0.5/90	1.3/151

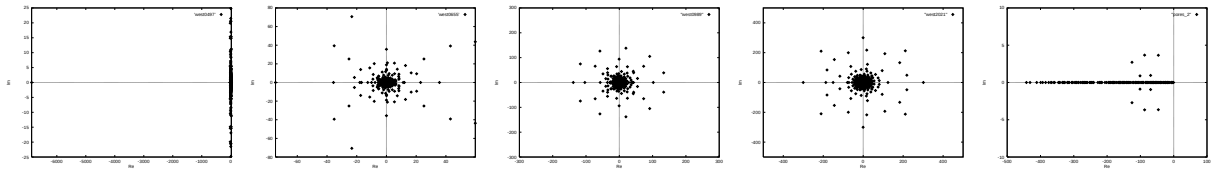

FIG. 1. *Computed spectra of WEST0497, WEST0655, WEST0989, WEST2021, and PORES2*

5. Parallelization of QR algorithm. The above results on the complexity of our method indicate the necessity of more efficient computation of the Arnoldi iteration. Although the speed of convergence 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 the subspace dimension m.

To avoid QR to become a bottleneck, we propose here a new data mapping method and a schedule of the computation for the parallel Hessenberg double shifted QR algorithm on distributed memory processors. Figure 2 shows the data mapping, where the number of the processors $p = 6$. This method is based on the partition of the matrix into $2p \times 2p$ blocks. The mapping is similar to the block Hankel-wrapped storage scheme in that the matrix is partitioned into $2p$ *strips* along the subdiagonal, and that each processor owns two strips at an interval of p. However, the strips are shifted left by 1.5 blocks, and this shift makes the loads near the diagonal so light that the lookahead step can be executed at the same time with the updates of the previous block transformation. We use a 'half block' as a unit of computation: We assume that each computation of the lookahead step and the column rotations of a diagonal block, whose nonzero elements are about a half of a block, is a half block. The time taken to execute the computation of a half block is a 'quarter', because each processor has four half blocks of computations in a block transformation.

Figure 2 also shows the schedule of the computations in the fourth block transformation. Each processor has four half blocks of computations and the order of the computations is shown with the number 1 to 4. The arrows depict the required communication. The long arrows from the diagonal block stand for the broadcast of the transformations. The lookahead step is executed by the processor 5 in the third quarter. Therefore, there is time of a quarter from the end of a lookahead step to the beginning of the transformations that use the results of the lookahead step, and it becomes possible to hide the latency of the broadcast of the transformations. The column rotation of the diagonal block was done in the first quarter. The row rotations in a processor are executed from right to left and the column rotations in a processor are executed from bottom to top, because the results of the half blocks at the right and the bottom must be sent to the next processors. With this ordering, at least two quarters of time are available to hide the latency of each communication.

The graph in Figure 2 shows the parallel performance of our program without matrix size reduction on a Fujitsu AP1000+, a distributed memory multicomputer system with 256 SuperSparc10 processors (50 MHz). The graph shows the relation between Mflops per processor and n/p with several values for p. The peak performance of the Hessenberg double shift QR algorithm on a single processor of AP1000+ is about 20.8 MFlops, using unrolling and tiling. Therefore, the parallel efficiency of 50% is attained with $n/p < 40$, and the parallel efficiency becomes 90% with $n/p \approx 150$. Such high parallel efficiency has rarely been observed in preceding researches on the parallel double shifted QR algorithm [1].

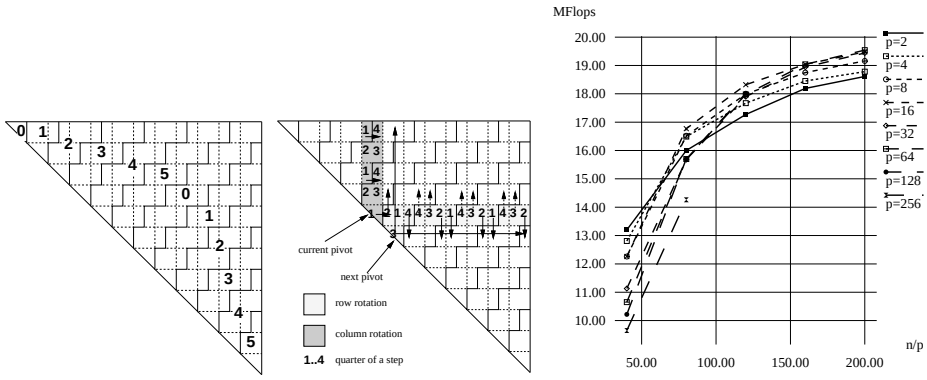

FIG. 2. *The proposed data mapping method and Mflops per processor vs n/p for first iterations. The broken lines in the left figures indicate the boundaries of the blocks, and the solid lines show the boundaries of the elements allocated to different processors. The numbers indicate to which processor each region should be allocated.*

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. Although the validity of our method was confirmed by numerical experiments, 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. Our new data mapping for double shifted QR algorithm, in which the loads including the lookahead step are balanced and the computations are pipelined by hiding the communication latency, is to become a promising method for the problem. The integration of these two approaches is the current problem.

REFERENCES

[1] G. HENRY AND R. VAN DE GEIJN, *Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problems: myth and reality*, SIAM. J. Sci. Comput., 17 (1996), pp. 870–883.

[2] 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.

[3] A. NISHIDA, R. SUDA, AND Y. OYANAGI, *Polynomial Acceleration for Restarted Arnoldi Method and its Parallelization*, in Proceedings of Third IMACS International Symposium on Iterative Methods in Scientific Computation, USA, to appear.

[4] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Wiley, New York, 1992.