Polynomial Acceleration for Restarted Arnoldi Iteration and its Parallelization

Akira Nishida Reiji Suda Yoshio Oyanagi

Abstract

We propose an accelerating method for the restarted Arnoldi iteration to compute a number of eigenvalues of the standard eigenproblem $Ax = \lambda x$ and discuss the dependence of the convergence rate of the accelerated iteration on the distribution of spectrum. The effectiveness of the approach is proved by numerical results. We also 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.

KEYWORDS: Arnoldi's method, preconditioning techniques, parallel QR algorithm

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 by Saad [5] with the explicitly restarting technique (ERA), by which the dimensions of the Krylov subspaces is kept modest. Although the restarted Arnoldi iteration is a quite effective approach, the dimension of the subspace becomes inevitably large, especially when the wanted eigenvalues are clustered. In this paper, we propose a simplified least-squares based method to accelerate the convergence of the restarted Arnoldi iteration.

The algorithm of the explicitly restarted Arnoldi iteration is summarized in Table 1. The choice of subspace dimension m is usually a tradeoff between the length of the iteration and the rate of convergence. Suppose $A \in \mathbf{R}^{n \times n}$ is diagonalizable with eigensolutions (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 we have $\psi(A)x_1 = u_1\psi(\lambda_1)\zeta_1 + \cdots + u_n\psi(\lambda_n)\zeta_n$. Assuming that the eigenvalues are ordered so that the wanted k ones are at the beginning of the expansion, we seek a polynomial such that $\max_{i=k+1,...,n} |\psi(\lambda_i)| < \min_{i=1,...,k} |\psi(\lambda_i)|$. The acceleration techniques and hybrid methods presented by Saad [5] attempt to improve the explicitly restarted Arnoldi iteration by approximately solving this min-max problem. 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 choice of ellipses as enclosing regions in

put here title
put here (eds.), pp. 1–5.

put here ISBN

Table 1. Block version of explicitly restarted Arnoldi iteration 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
       $H_{i,j} = V_i^T W_j; \quad W_j = W_j - V_i H_{i,j}$
     end for
     $Q_j R_j = W_j; \quad V_{j+1} = Q_j; \quad H_{j+1,j} = R_j$
   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 $\psi_k(\lambda)$ of degree $k$ by $\mathrm{Sp}(H_m) - \{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$.
6. $\tilde{X}_k = \psi_k(A)\tilde{X}_0; \quad Q_k R_k = \tilde{X}_k; \quad V_1 = Q_k$
7. Goto 2.
```

Chebyshev acceleration, however, may be overly restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance to 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 w. The only restriction with this technique is that the degree of the polynomial is limited because of cost and storage requirements. This can be overcome by compounding low degree polynomials, and the stability of the computation is enhanced by employing a Chebyshev basis.

2 Orthogonality based method

In this paper, we propose a simple method for determining the least squares polynomials which minimize the L_2 norm, defined on the boundary of the convex hull. By the maximum principle, the maximum modulus of $|\psi_n(\lambda)|$ is found on the boundary of some region H of the complex plane that includes the spectrum of A. We define an accelerating polynomial by a least squares residual polynomial minimizing the L_2 norm with respect to some weight $w(\lambda)$ on the boundary of H. Suppose that the $\mu + 1$ points $h_0, h_1, \cdots, h_\mu$ constitute the vertices of H. On each edge $h_{\nu-1}h_\nu$, $\nu = 1, \cdots, \mu$, 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 $\langle p, q\rangle = \sum_{\nu=1}^{\mu} \int_{h_{\nu-1}h_\nu} p(\lambda)\overline{q(\lambda)} w_\nu(\lambda)|d\lambda|$. 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.

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 $C^2 + \sum_{i=0}^{n-1} c_i^2$, which assumes its minimum at $c_0 = c_1 = \cdots = c_{n-1} = 0$. 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 ortho-normality condition. Note that the orthonormality of ψ_0 and ψ_1 must hold and each expansion of $\psi_i(\lambda)$ at each edge needs to be consistent [4]. 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 2 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 3 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 3, 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 [4] 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 2. 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

3 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

Table 3. 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

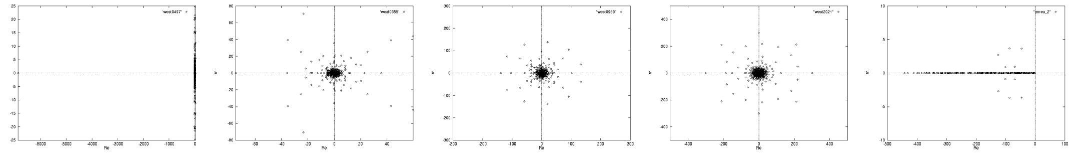

Figure 1. Computed spectra of WEST0497, WEST0655, WEST0989, WEST2021, and PORES2

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].

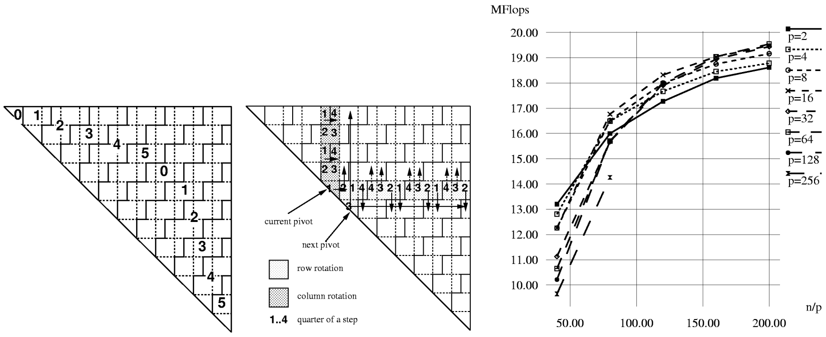

Figure 2. The proposed data mapping method and Mflops per processor vs n/p for first iterations. The broken lines in the left figures indicate the boundaries of the blocks, and the solid lines show the boundaries of the elements allocated to different processors. The numbers indicate to which processor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

4 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] Henry, G. and van de Geijn, R., Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problems: myth and reality. *SIAM. J. Sci. Comput.*, 17(4):870–883, 1996.

[2] Lehoucq, R. B. and Scott, J. A., An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices. Technical Report MCS-P547-1195, Argonne National Laboratory.

[3] Manteuffel, T. A., The Tchebychev iteration for nonsymmetric linear systems. *Numer. Math.*, 28:307–327, 1977.

[4] Nishida, A. and Oyanagi, Y., A Polynomial Acceleration of the Projection Method for Large Nonsymmetric Eigenvalue Problems. 1996 SIAM Annual Meeting, USA, Jul. 1996.

[5] Saad, Y., *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press, Manchester, 1992.

Akira Nishida, Reiji Suda, and Yoshio Oyanagi
Department of Information Science
University of Tokyo
Tokyo, 113 JAPAN
{nishida, reiji, oyanagi}@is.s.u-tokyo.ac.jp