INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 18, Number 5, Pages 712–721

EXTRAPOLATING THE ARNOLDI ALGORITHM TO IMPROVE EIGENVECTOR CONVERGENCE

SARA POLLOCK AND L. RIDGWAY SCOTT

Abstract. We consider extrapolation of the Arnoldi algorithm to accelerate computation of the dominant eigenvalue/eigenvector pair. The basic algorithm uses sequences of Krylov vectors to form a small eigenproblem which is solved exactly. The two dominant eigenvectors output from consecutive Arnoldi steps are then recombined to form an extrapolated iterate, and this accelerated iterate is used to restart the next Arnoldi process. We present numerical results testing the algorithm on a variety of cases and find on most examples it substantially improves the performance of restarted Arnoldi. The extrapolation is a simple post-processing step which has minimal computational cost.

Key words. Eigenvalue computation, extrapolation, Arnoldi algorithm.

1. Introduction

There are many applications in which the smallest eigenvalues of large systems must be computed, e.g., in stability analysis of numerical schemes [6] and in stability analysis of partial differential equations [16]. The (inverse) power method is often preferred due to its ease of implementation and the limited amount of storage required. In many cases [16], all that is required to implement the inverse power method is to solve the associated system of equations repeatedly. Thus it is easy to modify a code for a solver to become a code for the inverse power method.

Extrapolation has been shown [11] to provide an effective way to improve the power method. Here we consider a different approach in which the power method is first generalized. We then examine extrapolation of the method.

One generalization of the power method is to use a small number $k > 1$ of approximation vectors and to project the eigenproblem onto the corresponding k-dimensional space. One then solves the projected k-dimensional eigenproblem and extracts the eigenvector corresponding to the extreme eigenvalue as the next iterate. This can lead to faster convergence with a controlled increase in storage. A natural set of vectors to use is a Krylov basis, and we dub this approach the k-step Krylov method. We show that the popular LOBPCG method is of this form. There could of course be other ways of generating appropriate vectors at each step, e.g., a random process.

We show that the Arnoldi algorithm provides a very stable implementation of the k-step Krylov method. We further demonstrate how a simple extrapolation technique, which takes a combination of the two latest Arnoldi outputs as the next approximation, can be used to further enhance the rate of convergence. Since the basic k-step method is Arnoldi, it is remarkable that this algorithm can be improved by extrapolation.

Received by the editors January 28, 2021 and, in revised form, July 5, 2021.
2000 *Mathematics Subject Classification.* 65B05,65F15.

2. k-step Krylov methods

We can define general k-step Krylov methods as follows. We start with a vector $\mathbf{y}_1$ and define $\mathbf{y}_{j+1} = A\mathbf{y}_j$, for $j = 1, \dots, k-1$. We seek to find coefficients $a_1, \dots, a_k$ such that

$$A \sum_j a_j \mathbf{y}_j = \lambda \sum_j a_j \mathbf{y}_j. \tag{1}$$

Taking dot products, we see that this is equivalent to

$$K\mathbf{a} = \lambda M\mathbf{a},$$

where

$$K_{ij} = \mathbf{y}_i^t A \mathbf{y}_j, \qquad M_{ij} = \mathbf{y}_i^t \mathbf{y}_j.$$

Thus we can determine $\mathbf{a}$ by solving the eigenproblem

$$M^{-1}K\mathbf{a} = \lambda\mathbf{a}. \tag{2}$$

Due to the close connection with Ritz methods, $\mathbf{a}$ is called the Ritz vector and λ is the Ritz value. A great deal is known about how these approximate eigenvectors and eigenvalues for A as k increases [14].

We define the output of the k-step method as $\lambda_1, \dots \lambda_m$ $(m < k)$ and

$$\mathbf{y} = \sum_{i=1}^{k} a_i \mathbf{y}_i,$$

where $\mathbf{a}$ is the eigenvector corresponding to the extreme eigenvalue λ_1 for (2), and $\lambda_2, \dots \lambda_m$ are the remaining eigenvalues in descending order.

The eigenproblem (2) can be solved analytically for $k \leq 4$. In [13], the case $k = 2$ is described in detail.

2.1. Orthogonalization of Krylov vectors.

Unfortunately, the naive approach (2) to the k-step method fails for larger k, due to ill conditioning, as described in [13]. Thus we consider orthogonalization of the Krylov vectors.

Now we modify the steps leading to (2). We start with a vector $\hat{\mathbf{y}}_1$ and define Krylov vectors $\hat{\mathbf{y}}_{j+1} = A\hat{\mathbf{y}}_j$, for $j = 1, \dots, k$. Then we orthogonalize to get $\mathbf{y}_1, \dots, \mathbf{y}_k$ by the modified Gram–Schmidt algorithm [2, 12]. We provide the details in [13]. In this setting, the matrix M is the identity.

The use of orthogonal Krylov vectors allows extension to more steps k, but for slightly larger k the algorithm still fails due to the increasing condition number of K, as indicated in [13].

2.2. Arnoldi algorithm.

The Arnoldi algorithm makes a small change in the order of orthogonalization and multiplication by the matrix A. Instead of first creating the Krylov vectors all at once, we multiply by A only after orthogonalization. Thus $\mathbf{y}_1 = \|\hat{\mathbf{y}}_1\|^{-1}\hat{\mathbf{y}}_1$. Then for $n = 1, 2, \dots, k-1$, define

$$\begin{gathered} \widetilde{\mathbf{y}}_n = A\mathbf{y}_n - \sum_{j=1}^{n} h_{j,n}\mathbf{y}_j\,, \qquad h_{j,n} = \mathbf{y}_j^{\,t} A\mathbf{y}_n\,, \quad j = 1, \dots n, \\ h_{n+1,n} = \|\widetilde{\mathbf{y}}_n\|, \qquad \mathbf{y}_{n+1} = h_{n+1,n}^{-1}\widetilde{\mathbf{y}}_n. \end{gathered} \tag{3}$$

For $n = k$, we compute $h_{j,k} = \mathbf{y}_j^{\,t} A\mathbf{y}_k$ for $j = 1, \dots k$, but we do not perform the orthogonalization steps in the first line of (3) for $n = k$.

One can show by induction that the vectors $\mathbf{y}_j$ are orthogonal, so that, in exact arithmetic, $H = K$. But this subtle change makes the algorithm far more robust, as shown in Figure 1(a). The k-step algorithm approximates accurately many of the