J. KSIAM **Vol.3**, No.2, 5-16, 1999

A PROJECTION ALGORITHM FOR SYMMETRIC EIGENVALUE PROBLEMS

PIL SEONG PARK

Abstract

We introduce a new projector for accelerating convergence of a symmetric eigenvalue problem $A\mathbf{x} = \mathbf{x}$, and devise a power/Lanczos hybrid algorithm. Acceleration can be achieved by removing the hard-to-annihilate nonsolution eigencomponents corresponding to the widespread eigenvalues with modulus close to 1, by estimating them accurately using the Lanczos method. However, the additional Lanczos results can be obtained without expensive matrix-vector multiplications but a very small amount of extra work, by utilizing simple power-Lanczos interconversion algorithms suggested. Numerical experiments are given at the end.

1. Introduction

Numerical models often yield eigenvalue problems $A\mathbf{x} = \lambda\mathbf{x}$ for finding the dominant eigenvector $\mathbf{x}$ corresponding to the eigenvalue λ with the largest modulus. In many cases, the dominant eigenvalue is known in advance. One such example is the queuing problem $Q\mathbf{x} = 0$ described in [2], that can be converted to an eigenvalue problem $A\mathbf{x} = \mathbf{x}$ for finding the dominant eigenvector corresponding to the eigenvalue 1.

It is well known that, if the moduli of some eigenvalues of A are nearly equal to that of the dominant one we look for, usual algorithms such as the power method or its variants like the Chebyshev iteration do not work well, since the convergence depends on the modulus ratio of the second largest to the dominant. To improve the convergence in such cases, an orthogonal projector was proposed in [3] under the assumption that these unwanted eigenvalues are clustered closely to each other. However the projector sometimes may not work well if these unwanted major eigenvalues are well separated.

In this paper, we introduce a better orthogonal projector to deal with such cases, and devise a new power/Lanczos hybrid algorithm for symmetric eigenvalue problems. Numerical results of the algorithm in various cases are given at the end.

Throughout this paper, we deal with an $n \times n$ real symmetric eigenvalue problem $A\mathbf{x} = \mathbf{x}$, i.e., we look for the eigenvector corresponding to the dominant eigenvalue 1. If the dominant eigenvalue, theoretically known in advance, is different from 1, we can scale the problem to satisfy this condition.

Key words: projector, Lanczos method, the power method, power-Lanczos interconversion, symmetric eigenvalue problems, Krylov subspace

2. Some backgrounds

We first review the method introduced in [3] and look at the Lanczos method for further development.

Definition 1 *Let $(\lambda_j, \mathbf{z}_j)$ be the jth eigenpair of a given matrix A numbered in decreasing order by its eigenvalue modulus, and $\mathbf{z}_1$ be the eigenvector we look for, corresponding to $\lambda_1 = 1$.*

Definition 2 *For any vector $\mathbf{x}$, we define the residual of $\mathbf{x}$ by $\mathbf{r} = (A - I)\mathbf{x}$.*

The "unnormalized" power method $\mathbf{x}_{i+1} := A\mathbf{x}_i$ is one of the main driving force. Then the residual is just the difference between two consecutive power iterates and can be computed without extra matrix-vector multiplication. The solution component $\mathbf{z}_1$ in power iterates does not change, and residuals contain nonsolution components only, but not $\mathbf{z}_1$. Hence convergence of an iterate can be estimated by normalizing it and then computing the norm of its residual.

Let $\mathbf{x} = \sum_{j=1}^{n} \beta_j \mathbf{z}_j$ be the initial vector for the unnormalized power iteration. After enough(say, e) power iterations, let $\mathbf{x}_1$ be the first power iterate of our concern and $\mathbf{r}_1$ be its residual. Then

$$\mathbf{x}_1 = A^e\mathbf{x} = \sum_{j=1}^{n} \beta_j A^e \mathbf{z}_j = \sum_{j=1}^{n} \beta_j {\lambda_j}^e \mathbf{z}_j.$$

Assume that the nonsolution components in $\mathbf{x}_1$ mainly consist of k major components(say, major nonsolution components) $\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{k+1}$, and others(say, minor nonsolution components) are negligibly small, i.e., $1 \gg |\beta_j {\lambda_j}^e| \gg |\beta_i {\lambda_i}^e|$, $j = 2, \ldots, k+1$, $i = k+2, \ldots, n$. If we could remove most of these k major components, we get very fast convergence.

Note that, after enough power iterations, residuals will be rich in a few major nonsolution components, depending on the eigenstructure of A. Hence residuals are used to approximate the subspace spanned by these major nonsolution components, and an orthogonal projector was designed to reduce them in power iterates as described below.

We apply $k+1$ more power iterations to obtain $\mathbf{x}_{i+1} = A^i\mathbf{x}_1, i = 1, 2, \ldots, k+1$, and compute the residuals $\mathbf{r}_i \equiv \mathbf{x}_{i+1} - \mathbf{x}_i$, $i = 1, 2, \ldots, k+1$. By applying the Gram-Schmidt process to the k residuals $\mathbf{r}_1, \ldots, \mathbf{r}_k$, we form the matrix $V \in \mathbf{R}^{n \times k}$ whose columns are orthonormal and $\mathcal{R}(V) = \text{span}\{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$, which may be close to $\text{span}\{\mathbf{z}_2, \ldots, \mathbf{z}_{k+1}\}$. By subtracting the nonsolution components in $\mathbf{r}_{k+1}$ projected onto $\mathcal{R}(V)$, most of the major nonsolution components in the iterate $\mathbf{x}_{k+1}$ can be removed effectively by the projection step

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{k+1} + \frac{1}{1-\mu} V V^T \mathbf{r}_{k+1}, \tag{1}$$

where μ is an approximation to the eigenvalues corresponding to the major eigencomponents $\mathbf{z}_2, \ldots, \mathbf{z}_{k+1}$ we try to remove. Note that $\mathbf{x}_{k+2}$ is needed only for the computation of $\mathbf{r}_{k+1}$ and is not used any further.

In an ideal case when $\mathcal{R}(V) = \text{span}\{\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{k+1}\}$,

Theorem 1 *Let $V \in \mathbf{R}^{n\times k}$ be the matrix with orthonormal columns such that*

$$\mathcal{R}(V) = \text{span}\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k\} = \text{span}\{\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{k+1}\}.$$

Let the current iterate be $\mathbf{x}_{k+1} = \sum_{j=1}^{n} \alpha_j \mathbf{z}_j$, where $|\alpha_1| = O(1)$, $|\alpha_i| \le \epsilon_1$ for $i = 2, \ldots, k+1$, and $|\alpha_j| \le \epsilon_2$ for $j = k+2, \ldots, n$, and $\epsilon_2 \ll \epsilon_1 \ll 1$. If μ is chosen so that $|\frac{\lambda_j - \mu}{1-\mu}| \le \delta$ for some $0 < \delta \ll 1$ for $j = 2, \ldots, k+1$, then the remaining major nonsolution components $\mathbf{z}_2, \ldots, \mathbf{z}_{k+1}$ after the projection step (1) are at most $O(\epsilon_k)$ where $\epsilon_k = \max(k\delta\epsilon_1, t\epsilon_2)$ and $t = (n-k-1)(1 + \frac{2}{1-\mu})$.

The Lanczos method

Given a matrix $A \in \mathbf{R}^{n\times n}$ and a set $\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ of k linearly independent vectors($k \ll n$), the projection method on $\text{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ tries to approximate an eigenpair $(\lambda, \mathbf{z})$ of the matrix A by a pair $(\lambda^{(k)}, \mathbf{z}^{(k)})$ satisfying [5]

$$\begin{aligned} &\mathbf{z}^{(k)} \in \text{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}, \\ &(A - \lambda^{(k)} I)\mathbf{z}^{(k)} \perp \mathbf{q}_j, \qquad j = 1, 2, \ldots, k. \end{aligned}$$

The solutions $\lambda^{(k)}$ are called *Ritz values* on the subspace $\text{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$, and to each Ritz value is associated a *Ritz vector* $\mathbf{z}^{(k)}$ [7].

Let $Q_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$. Writing $\mathbf{z}^{(k)} = Q_k \mathbf{s}^{(k)}$, we see that $(\lambda^{(k)}, \mathbf{z}^{(k)})$ are eigenpairs of the problem

$$(T_k - \lambda^{(k)} B_k)\mathbf{s}^{(k)} = 0,$$

or equivalently

$$(B_k^{-1} T_k - \lambda^{(k)} I)\mathbf{s}^{(k)} = 0,$$

where $T_k = Q_k^T A Q_k$ and $B_k = Q_k^T Q_k$.

In usual applications, we choose an orthonormal system Q_k, so that B_k reduces to the identity matrix. One such process is the symmetric Lanczos method. It uses the orthonormal system obtained by orthogonalization of the *Krylov vectors* $\mathbf{q}_1, A\mathbf{q}_1, \ldots, A^{k-1}\mathbf{q}_1$, where $\mathbf{q}_1$ is a starting vector. Then the matrix T_k becomes tridiagonal, and letting

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix},$$

the entries are easily obtainable from a three-term recurrence relation [7]

$$A\mathbf{q}_j = \beta_{j-1}\mathbf{q}_{j-1} + \alpha_j \mathbf{q}_j + \beta_j \mathbf{q}_{j+1}, \quad j = 1, \ldots, n-1,$$

$$\beta_0 \mathbf{q}_0 \equiv 0.$$

3. A new projector

The reduction/magnification factor of the jth eigencomponent $\mathbf{z}_j$ by the projection step (1) in the previous method is $|\lambda_j - \mu|/|1 - \mu|$. One major drawback of the previous algorithm is that nonsolution components can grow if this ratio is larger than 1, especially if the eigenvalue λ_j is far away from the point $(\mu, 0)$ in the complex plane.

The parameter μ in the old projector (1) is a representative value for the k eigenvalues that correspond to the major nonsolution components, which are different in general. Moreover if these eigenvalues are well-separated, a projection step using one parameter can be a disaster.

To remedy such phenomena, we modify the old projector to a multi-parametered one. That is, we replace the old orthogonal projector (1) by

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{k+1} + V\Lambda V^T \mathbf{r}_{k+1}, \tag{2}$$

where columns of $V \in \mathbf{R}^{n\times k}$ are orthonormal, and $\mathcal{R}(V) = \text{span}\{\mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_{k+1}\}$ and

$$\Lambda = \text{diag}(\frac{1}{1-\mu_2}, \ldots, \frac{1}{1-\mu_{k+1}})$$

where μ_j's are hopefully close to λ_j's.

As before, after enough power iterations, assume that the current residual $\mathbf{r}_{k+1}$ mainly consists of k major nonsolution components $\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{k+1}$, and the rest are much smaller. Now consider the new projection step (2). In an ideal case when the columns of V are exactly $\mathbf{z}_2, \ldots, \mathbf{z}_{k+1}$ and $\mu_j = \lambda_j$, $j = 2, \ldots, k+1$, we have

Theorem 2 *Let $V \in \mathbf{R}^{n\times k}$ be the matrix with orthonormal columns such that $V = [\mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_{k+1}]$, and let $\mu_j = \lambda_j$, $j = 2, 3, \ldots, k+1$. After enough power iterations, assume that the current iterate can be written as $\mathbf{x}_{k+1} = \sum_{j=1}^{n} \alpha_j \mathbf{z}_j$, where $|\alpha_1| = O(1)$, $|\alpha_i| \le \epsilon_1$ for $i = 2, \ldots, k+1$, and $|\alpha_j| \le \epsilon_2$ for $j = k+2, \ldots, n$, and $\epsilon_2 \ll \epsilon_1 \ll 1$. Then after applying the projection step (2), the sum of the remaining nonsolution components is at most of $O(n\epsilon_2)$.*

Proof. Let $V = [\mathbf{z}_2, \ldots, \mathbf{z}_{k+1}]$ and $W = [\mathbf{z}_{k+2}, \ldots, \mathbf{z}_n]$. Then both V and W have orthonormal columns since A is symmetric. We can write the current iterate $\mathbf{x}_{k+1}$ and its residual $\mathbf{r}_{k+1}$ in a matrix form as $\mathbf{x}_{k+1} = \alpha_1\mathbf{z}_1 + V\zeta_2 + W\zeta_3$ and $\mathbf{r}_{k+1} = (A-I)\mathbf{x}_{k+1} = \sum_{j=2}^{n} \alpha_j(\lambda_j - 1)\mathbf{z}_j = V\xi_2 + W\xi_3$ where $\zeta_2 = [\alpha_2, \ldots, \alpha_{k+1}]^T$, $\zeta_3 = [\alpha_{k+2}, \ldots, \alpha_n]^T$, $\xi_2 = [\alpha_2(\lambda_2 - 1), \ldots, \alpha_{k+1}(\lambda_{k+1} - 1)]^T$, and $\xi_3 = [\alpha_{k+2}(\lambda_{k+2} - 1), \ldots, \alpha_n(\lambda_n - 1)]^T$.

Hence after the projection step, we obtain

$$\begin{aligned}
\mathbf{x}_{\text{new}} &= \mathbf{x}_{k+1} + V\Lambda V^T \mathbf{r}_{k+1} \\
&= \alpha_1\mathbf{z}_1 + V\zeta_2 + W\zeta_3 + V\Lambda V^T(V\xi_2 + W\xi_3) \\
&= \alpha_1\mathbf{z}_1 + V\zeta_2 + W\zeta_3 + V\Lambda\xi_2 \\
&= \alpha_1\mathbf{z}_1 + W\zeta_3
\end{aligned}$$

since $V^TV = I$, $V^TW\xi_3 = 0$, and $\Lambda\xi_2 = -\zeta_2$. Hence the size of the sum of nonsolution components in $\mathbf{x}_{\text{new}}$ is

$$||\mathbf{x}_{\text{new}} - \alpha_1\mathbf{z}_1|| = ||\sum_{j=k+2}^{n} \alpha_j\mathbf{z}_j|| \leq n\epsilon_2.$$

Note that we need compute the exact eigenpairs $(\lambda_j, \mathbf{z}_j)$, $j = 2, \ldots, k+1$ for better convergence. The Lanczos method, which is known to compute a few extreme eigenpairs very accurately, is a good candidate for this purpose.

Hence in our new algorithm, we make use of the Ritz pairs(e.g., see [1]). That is, by applying the Lanczos method to the residual $\mathbf{r}_1$, we compute k Lanczos vectors $\mathbf{q}_1, \ldots, \mathbf{q}_k \in \mathbf{R}^n$ and a $k \times k$ tridiagonal matrix T_k. Through eigenanalysis of T_k, we compute its eigenpairs $(\theta_j, \mathbf{s}_j)$, $j = 2, \ldots, k+1$ (We intentionally number eigenvalues in this way to conform future development.). It is well known that a few of θ_j's are good approximations to the extreme eigenvalues $\lambda_2, \lambda_3, \ldots$ of A(since the Lanczos method is applied to the residual $\mathbf{r}_1$), and some of $Q_k\mathbf{s}_j$'s where $Q_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$, are good approximations to the eigenvectors $\mathbf{z}_2$,$\mathbf{z}_3, \ldots$, of A.

Hence we select only c eigenpairs $(\theta_j, \mathbf{s}_j)$, $j = 2, \ldots, c+1 (c < k$, that are believed to be accurate), out of them and obtain $\Lambda \in \mathbf{R}^{c\times c}$ and c approximations $\mathbf{y}_j = Q_k\mathbf{s}_j$, $j = 2, \ldots, c+1$, to the eigenvectors $\mathbf{z}_2, \ldots, \mathbf{z}_{c+1}$ of A. Then take $V = [\mathbf{y}_2, \ldots, \mathbf{y}_{c+1}] \in \mathbf{R}^{n\times c}$.

4. Interconversion between the power iterates and the Lanczos results

It seems that, according to the previous explanation, we need apply the Lanczos method too, independently of the power method, that requires a lot of extra costly matrix-vector multiplications. However we can avoid it by carefully considering the relation among power iterates, their residuals, and the Lanczos vectors computed from the residuals.

The reason for such interconversion is because the power iterates, their residuals, and the Lanczos method all make use of Krylov subspaces, and the residual has been defined so that such interconversion is possible.

Lemma 1 *Consider the power iteration $\mathbf{x}_{i+1} := A\mathbf{x}_i$ for an eigenvalue problem $A\mathbf{x} = \mathbf{x}$ with an initial vector $\mathbf{x}_1$. Then, not only the power iterates $\mathbf{x}_1, \mathbf{x}_2, \ldots$ but also their residuals $\mathbf{r}_1, \mathbf{r}_2, \ldots$ form Krylov subspaces.*

Proof. The facts are clear by the definition of the Krylov subspace and Definition 2.

This means that, the residual sequence $\mathbf{r}_1, \mathbf{r}_2, \ldots$ can be thought of as another power iterates by $\mathbf{r}_{i+1} := A\mathbf{r}_i$. Power iterates can be constructed using the residual sequence and the initial iterate $\mathbf{x}_1$ by

Lemma 2 *Let $\mathbf{x}_1, \mathbf{x}_2, \ldots$ be power iterates. Then $\mathbf{x}_{j+1} = \mathbf{x}_1 + \sum_{i=1}^{j} \mathbf{r}_i$.*

Getting Lanczos tridiagonal matrix from power iterates

Since we apply the Lanczos method to the residual $\mathbf{r}_1$,

Theorem 3 *The Lanczos tridiagonal matrix T_k can be obtained from residual iterates $\mathbf{r}_1$,$\mathbf{r}_2$,...,$\mathbf{r}_k$ with an additional matrix-vector multiplication and some inner products.*

Proof. It is just a symmetric version of the Theorem 2 in [4], with power iterates replaced by the residual iterates.

Corollary 1 *The Lanczos tridiagonal matrix T_k can be constructed from power iterates $\mathbf{x}_1$,$\mathbf{x}_2$,...,$\mathbf{x}_{k+2}$.*

Getting power iterates from Lanczos process

Conversely, we can obtain power iterates from Lanczos results.

Theorem 4 *The residual sequence $\mathbf{r}_2, \ldots, \mathbf{r}_k$ can be computed using the Lanczos results by*

$$\mathbf{r}_j = A^{j-1}\mathbf{r}_1 = ||\mathbf{r}_1|| V T_k^{j-2} \mathbf{t}_1$$

where $V \in \mathbf{R}^{n\times k}, T_k \in \mathbf{R}^{k\times k}$, and $\mathbf{t}_1$ is the first column of the Lanczos tridiagonal matrix T_k.

Proof. Consider the full Lanczos' relation

$$AV = VT, \tag{3}$$

where $A, V, T \in \mathbf{R}^{n\times n}$. Let $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$. The first column of AV can be written as

$$A\mathbf{v}_1 = V\mathbf{t}_1 \tag{4}$$

where $\mathbf{t}_1$ is the first column of T. Premultiplying (4) by A and using (3),

$$A^2\mathbf{v}_1 = AV\mathbf{t}_1 = VT\mathbf{t}_1.$$

In general, we obtain

$$A^j\mathbf{v}_1 = VT^{j-1}\mathbf{t}_1.$$

We only need the first 2 columns of T^{j-1} to compute $A^j\mathbf{v}_1$, since T is tridiagonal and only the first 2 entries of $\mathbf{t}_1$ are nonzero. Hence we need only the first k columns of V and $k \times k$ principal submatrix of T, which is T_k. Since we apply the Lanczos method to the residual $\mathbf{r}_1$(i.e., the starting vector is $\mathbf{v}_1 = \mathbf{r}_1/||\mathbf{r}_1||$), the result is immediate.

Corollary 2 *The power iterates $\mathbf{x}_2, \ldots, \mathbf{x}_{k+1}$ can be obtained from the Lanczos matrix T_k and the Lanczos vectors.*

5. A power/Lanczos hybrid algorithm

Our algorithm consists of one or more hybrid steps, each of which consists of some power iterations(so that residuals mainly contain some major nonsolution components only), Lanczos steps(to estimate major nonsolution eigenpairs), and a projection step(to remove those major nonsolution components).

More precisely, at the beginning of each hybrid iteration, we apply some(say, s) power iterations to the most recent iterate $\mathbf{x}_{\text{new}}$ and obtain $\mathbf{x}_1$. Then we compute the residual $\mathbf{r}_1 = \mathbf{x}_2 - \mathbf{x}_1$ where $\mathbf{x}_2 = A\mathbf{x}_1$. At this point, we can use either of the following two methods:

- Continue applying power iteration to generate $\mathbf{x}_3, \ldots, \mathbf{x}_{k+2}$ by $\mathbf{x}_{i+1} := A\mathbf{x}_i$, $i =$ 2,3,...,$k+1$, then convert them to create the Lanczos matrix and vectors by Theorem 3 and Corollary 1.

or

- Stop power iteration, and apply the Lanczos method to $\mathbf{r}_1$ to get the Lanczos matrix T_k and the Lanczos vectors. Then the residual $\mathbf{r}_k$(hence the power iterate $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ too) can be obtained by Theorem 4 and Corollary 2.

In any case, the Lanczos matrix T_k and the Lanczos vectors are used to approximate the major nonsolution eigencomponents $(\lambda_j, \mathbf{z}_j)$, $j = 2, \ldots, k+1$, as explained previously. However, since only some(say, c where $c < s$) of them are accurate, we choose only those c eigenpairs and project the most recent residual $\mathbf{r}_{k+1}$ onto the subspace spanned by the c eigenpairs and subtract the components from $\mathbf{x}_{k+1}$.

Based on the latter that looks simpler, we suggest the following power/Lanczos hybrid algorithm (m, s, k, c), where s is the number of power iterations to be applied before the Lanczos step, k is the size of the Lanczos tridiagonal matrix to be constructed, and c is the number of dimensions onto which a projection step is applied(i.e., the number of eigenpairs that are assumed to be accurate). Note that we may allow extra m power iterations to the initial guess $\mathbf{x}_{\text{init}}$ before the first hybrid iteration begins (outside of the hybrid loop).

Algorithm 1 : Power/Lanczos hybrid algorithm(m, s, k, c)
Given a matrix $A \in \mathbf{R}^{n\times n}$, assume $c < k \ll n$.
1. Take an initial guess $\mathbf{x}_{\text{init}}$.
2. Apply m power iterations to $\mathbf{x}_{\text{init}}$.
3. For $i = 1, 2, \ldots,$ until convergence, do

   1) Apply s power iterations to the most recent iterate to obtain $\mathbf{x}_1$. Then compute $\mathbf{r}_1 = \mathbf{x}_2 - \mathbf{x}_1$ where $\mathbf{x}_2 = A\mathbf{x}_1$.

   2) Apply the Lanczos method to $\mathbf{r}_1$ to obtain a tridiagonal matrix T_k and k Lanczos vectors $\mathbf{q}_1, \ldots, \mathbf{q}_k$.

   3) Compute the eigenpairs $(\theta_j, \mathbf{s}_j)$, $j = 2, \ldots, k+1$, of T_k, and select the major c pairs out of them.

   4) Form $\Lambda \in \mathbf{R}^{c\times c}$ and $V = [\mathbf{y}_2, \ldots, \mathbf{y}_{c+1}] \in \mathbf{R}^{n\times c}$ where $\mathbf{y}_j = Q_k\mathbf{s}_j$, $j = 2, \ldots, c+1$, and $Q_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$.

   5) Use the Lanczos results to form $\mathbf{x}_{k+1}$ and $\mathbf{r}_{k+1}$.

   6) Perform a projection step to obtain a new iterate $\mathbf{x}_{\text{new}} = \mathbf{x}_{k+1} + V\Lambda V^T\mathbf{r}_{k+1}$ and normalize $\mathbf{x}_{\text{new}}$.

6. Numerical experiments and discussion

We created the following two relatively hard sample problems $A\mathbf{x} = \mathbf{x}$ as follows: for each problem, we created a two-queue overflow queuing problem $Q\mathbf{x} = 0$ with parameter

quadruples (s_j, w_j, i_j, o_j), where s_j is the number of servers, w_j is the number of wait spaces, i_j is the mean arrival rate, and o_j is the mean departure rate in the jth queue [2]. We convert this into the corresponding eigenvalue problem $G\mathbf{x} = \mathbf{x}$ by Jacobi splitting (e.g., see [1]). Since G is 2-cyclic(hence the power iteration does not converge as is, e.g., see [6]), we slightly shift it by $B = (G + 0.01I)/1.01$ so that the resulting matrix is acyclic. ¿From this, we get an eigenvalue problem $A\mathbf{x} = \mathbf{x}$, where A is obtained by symmetrizing B by $(B + B^T)/2$ and scaling the resulting matrix by its spectral radius so that $\lambda_1 = 1$ is a simple eigenvalue. The eigenvalues, other than 1, of each of the matrices are well separated, and many of them have relatively large modulus close to 1.

Problem 1 The parameter quadruples used are $(5, 3, 10, 6)$ and $(5, 3, 3, 1)$ in each queue($j = 1, 2$) respectively. Hence the matrix size is 81. Eigenvalues other than 1 are, in decreasing order of magnitude, -0.98144, 0.97465, -0.95610, 0.94286, -0.92430, 0.89565, -0.87709, 0.86261, 0.84435, -0.84405, 0.83755, -0.82580, -0.81900, 0.80051, and the rest are between -0.8 and 0.8.

Problem 2 The parameter quadruples used are $(7, 12, 33, 7)$ and $(4, 15, 22, 5)$ in each queue($j = 1, 2$) respectively. Hence the matrix size is 400. 0.99608, -098048, 0.98004, 0.97843, -0.97655, 0.97326, -0.96051, -0.95891, 0.95794, -0.95374, 0.95273 are the eigenvalues with modulus greater than 0.95. There are 14 eigenvalues whose moduli are between 0.90 and 0.95, and 14 more between 0.85 and 0.80, others being between -0.85 and 0.85.

As an initial guess $\mathbf{x}_{\text{init}}$, we use a vector of all ones scaled by its 2-norm. One projection step needs one matrix-vector multiplication and some extra calculation for eigenanalysis of the Lanczos tridiagonal matrix T_k of size k, etc. However, since $k \ll n$, we will simply count the work for one projection step as one matrix-vector multiplication, ignoring any extra calculation (According to actual timing result, the assumption seems to be acceptable.).

Table 1 shows the number of matrix-vector multiplications used to reduce the norm of the residual to certain sizes($10^{-3}, 10^{-7}$, and 10^{-10}) for Problem 1, by various methods: pure power method, shifted power method with best shift 0.006223529(computed by eigenanalysis of A), the old projection method(denoted by "Old") in [3], and the new hybrid algorithm with various parameters.

In general, regardless of the choice of parameters, the new hybrid algorithm worked well, reducing the amount of work to nearly 1/5 of the underlying power method and less than 1/2 of the old method. It also has been observed that, for fixed value of m, s, k, different choices of c do not give much difference. Anyhow, taking $c = k$ may sometimes give a slightly worse result since not all of the Lanczos pairs are accurate.

Fig. 1 shows a typical residual reduction pattern by the hybrid algorithm, together with those by the pure power method and the old projection method. We denote a specific algorithm by "algorithm (m, s, k, c)" where "algorithm" is either "old"(for the old projection method) or "hybrid"(for the new hybrid method) and m, s, k, c are the parameters as described previously. Note that rapid residual reduction after each

Table 1: The number of matrix-vector multiplications required to reduce the residual to certain sizes by various methods.

Algorithms	Parameters				Residual less than		
Projector	m	s	k	c	10^{-3}	10^{-7}	10^{-10}
Normal power iteration					89	555	924
Power iteration with best shift					82	441	731
Old	10	5	5	-	54	185	312
Hybrid				2	38	109	160
				3	36	109	160
				4	32	113	165
				5	32	111	154
Old	50	5	5	-	81	241	359
Hybrid				2	61	131	187
				3	61	116	168
				4	61	111	172
				5	61	116	163
Old	10	10	10	-	82	241	368
Hybrid				3	31	149	200
				5	31	94	139
				7	31	94	137
				9	31	94	137
Old	0	10	10	-	82	332	441
Hybrid				3	33	149	218
				5	25	86	140
				7	25	84	127
				9	25	84	127
Old	20	1	4	-	68	326	413
Hybrid				2	35	136	222
				3	34	137	232

Figure 1: A typical residual reduction pattern of the hybrid algorithm applied to problem 1.

projection is clearly visible. Comparing hybrid(10,10,10,5) and hybrid(0,10,10,5), we may conclude that it gives little difference whether we start a projection step slightly earlier or not, because the Lanczos method gives a very good approximation.

However for the result of old(10,10,10,-), the same values for m, s, k are used but rapid drop is seldomly seen(In fact, we did apply a projection step at every 20 power iterations, but most of them have been found of no use.). The reason seems to be that the estimation for major nonsolution component in the old algorithm is not so accurate as the Lanczos method used in the hybrid algorithm.

However, even though old(10,10,10, -) performs a projection step at every 21(i.e., 10 for power iterations, 11 for the Lanczos method applied to the residual) matrix-vector multiplications, rapid improvement is seldomly seen. In the figure, rapid improvement is seen at the 6th, 11th, and 15th projections only. Thus the old algorithm must have failed to accelerate convergence in most hybrid steps, unless more power iterations reduce the number of major nonsolution components so that the condition is favorable for the old projector.

When we apply projection steps too often(i.e., see the cases for $m = 20, s = 1, k = 4$ at the end of Table 1 and Figure 2), convergence may be worse, because it is not ready for another projection yet in the sense that the span of the residual sequence may not be a good approximation to the span of dominant nonsolution components. However, even in such cases, the new hybrid algorithm still works far better than the old algorithm.

Figure 3 shows that the new algorithm still works well even though the matrix of

Figure 2: Effect of frequent projections.(Problem 1).

Problem 2 has many more(compared to the size of the dimension $c = 5$) well-separated eigenvalues with modulus close to 1.

In general, any choice of the parameters (k, s, m, c) for the new hybrid algorithm seems to work well, but determination of their optimal values needs further research.

References

1. G. H. Golub and C. F. van Loan, 1996. *Matrix computations, 3rd Ed.,* Johns Hopkins Univ. Press, Baltimore, U. S. A.

2. L. Kaufman, 1983. *Matrix methods for queuing problems,* SIAM J. Sci. Comput., 4:525-552.

3. P. S. Park, 1996. *Use of an orthogonal projector for accelerating a queuing problem solver,* Korean J. Com. & Appl. Math. 3(2):193-204.

4. P. S. Park, 1997. *Interconversion between the power and Arnoldi's methods.* Comm. Korean Math. Soc. 12(1):145-155.

5. Y. Saad, 1980. *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices,* Lin. Alg. App., 34:pp. 269-295.

6. E. Seneta, 1981. *Non-negative matrices and Markov chains, 2nd Ed.,* Springer-Verlag, New York, U. S. A.

Figure 3: Residual reduction by the hybrid algorithm applied to problem 2.

7. L. N. Trefethen and D. Baus, III, 1997. *Numerical linear algebra,* SIAM, Philadelphia, U. S. A.

Department of Computer Science
University of Suwon
Kyungki-Do 445-743, Korea