[Applications and Applied Mathematics: An International Journal (AAM)](https://digitalcommons.pvamu.edu/aam)

Volume 5 | Issue 1 Article 13

6-2010

Developing an Improved Shift-and-Invert Arnoldi Method

H. Saberi Najafi
Guilan University

M. Shams Solary
Guilan University

Available at
http://pvamu.edu/aam
Appl. Appl. Math.
ISSN: 1932-9466

Vol. 5, Issue 1 (June 2010) pp. 167 - 180
(Previously, Vol. 5, No. 1)

Applications and Applied Mathematics:
An International Journal
(AAM)

Developing an Improved Shift-and-Invert Arnoldi Method

H. Saberi Najafi and M. Shams Solary
Department of Mathematics
Faculty of Sciences & Computer Center
Guilan University
P.O. Box 1914
Rasht, Iran
hnajafi@guilan.ac.ir
shamssolary@gmail.com

Received: May 6, 2009; Accepted: April 12, 2010

Abstract

An algorithm has been developed for finding a number of eigenvalues close to a given shift and in interval $[Lb, Ub]$ of a large unsymmetric matrix pair. The algorithm is based on the shift-and-invert Arnoldi with a block matrix method. The block matrix method is simple and it uses for obtaining the inverse matrix. This algorithm also accelerates the shift-and-invert Arnoldi Algorithm by selecting a suitable shift. We call this algorithm Block Shift-and-Invert or BSI. Numerical examples are presented and a comparison has been shown with the results obtained by Sptarn Algorithm in Matlab. The results show that the method works well.

Keywords: Eigenvalue, Shift-and-Invert, Arnoldi method, Inverse, Block matrix, LDV decomposition

MSC (2000) No.: 65N25, 65F25

1. Introduction

The eigenvalue problem is one of the most important subjects in Applied Sciences and Engineering. So this encouraged scientists into gaining new methods for this problem. For standard problems, powerful tools are available such as QR, Lanczos, Arnoldi Algorithm and etc. see Datta (1992) and Saad (1994). Computing the eigenvalues of the generalized eigenvalue problem ($Ax = \lambda Bx$) is one of the most important topics in numerical linear algebra.

The shift-and-invert Arnoldi method has been popularly used for computing a number of eigenvalues close to a given shift and/or the associated eigenvectors of a large unsymmetric matrix pair ($Ax = \lambda Bx$).

We consider the large unsymmetric generalized eigenproblem

$$A\phi_i = \lambda_i B\phi_i, \tag{*}$$

where A and B are $n \times n$ large matrices. An obvious approach is to transform (*) to a standard eigenproblem by inverting either A or B but if A or B are singular or ill-conditioned this manner will not work well. We are interested in computing some interior eigenvalues of (A, B) in the complex plane or some eigenvalues that are situated in the interval [Lb, Ub]. We will describe this problem and show a new method for solving this problem.

One of the most commonly used techniques for this kind of problem is the shift-and-invert Arnoldi method Jia and Zhang (2002), which is a natural generalization of the shift-and-invert Lanczos method for the symmetric case Ericsson and Ruhe (1980).

When $A - \sigma B$ is invertible for σ, the eigenvectors of the matrix pair (A, B) are the same as those of the matrix $(A - \sigma B)^{-1} B$. Therefore, we can run the Arnoldi method on the matrix $(A - \sigma B)^{-1} B$. If the shift σ is suitably selected, we set $C = (A - \sigma B)^{-1} B$ so the Arnoldi method applied to the eigenproblem of the shift- and- invert matrix C. It may give a much faster convergence with eigenvalues in interval including shift σ. Instead of a fixed or constant shift σ, Ruhe provided an effective technique Ruhe (1994) for selecting the shift σ dynamically, also can see Saberi and Shams (2005).

The shift-and-invert can be used when both A and B are singular or near singular. Since the shift-and-invert Arnoldi method for problem (*) is mathematically equivalent to the Arnoldi method for solving the transformed eigenproblem, the former has the same convergence problem as the latter; it is described in Section 2. This motivates us to derive a Block Shift-and-Invert Algorithm and to develop corresponding more efficient algorithms. In section 3, we will discuss on Block Shift-and-Invert Algorithm by block matrix and LDV decomposition. Then, we try to find eigenvalues for system $A\varphi_i = \lambda B\varphi_i$ in interval [*lb, ub*] by selecting a suitable shift. Section 4 describes Sptarn function in Matlab and some properties of it. Section 5 reports several numerical examples and compares Block Shift-and-Invert Algorithm with Sptarn Algorithm.

2. Shift-and-Invert

We start this section with a definition in generalized eigenvalue problem $A\varphi_i = \lambda_i B\varphi_i$ and then describe shift-and-invert method.

Definition 2.1:

In the generalized eigenvalue problem $(A\varphi_i = \lambda_i B\varphi_i)$, the matrix $(A - \lambda B)$ is called a matrix pencil. It is conveniently denoted by (A, B). The pair (A, B) is called regular if $\det(A - \lambda B)$ is not identically zero; otherwise, it is called singular.

We would like to construct linearly transformed pairs that have the same eigenvalues or eigenvectors as (A, B) and such that one of the two matrices in the pair is nonsingular. We have a theorem, see Golub and Van Loan (1989), Saad (1988) that shows when the pair (A, B) is a regular pair, then there are two scalars σ_*, τ_* such that the matrix $\tau_* A - \sigma_* B$ is nonsingular.

When one of the components of the pair (A, B) is nonsingular, there are simple ways that generalized problem transfer to a standard problem. For example $A\varphi_i = \lambda_i B\varphi_i \rightarrow B^{-1}A\varphi_i = \lambda_i \varphi_i$ or $BA^{-1}\varphi_i = \lambda_i \varphi_i$, or when A, B are both Hermitian and, in addition B is positive definite, we have $B = LL^T$ (Cholesky factorization) $L^{-1}AL^{-T}\varphi_i = \lambda_i \varphi_i$. None of the above transformations can be used when both A and B are singular. In this particular situation, a shift can help for solving the equation.

2.1. Reduction to Standard Form

We know that for any pair of scalars σ_1, σ_2 the pair $(A - \sigma_1 B, B - \sigma_2 A)$ has the same eigenvectors as the original pair (A, B) in $\beta Ax = \alpha Bx$ or $Ax = \dfrac{\alpha}{\beta} Bx$. An eigenvalue (α', β') of the transformed matrix pair is related to an eigenvalue pair (α, β) of the original matrix pair by $\alpha = \alpha' + \sigma_1 \beta'$, $\beta = \beta' + \sigma_2 \alpha'$.

Shift-and-invert for the generalized problems corresponds through two matrices (*A, B*), typically the first. Thus, the shift-and-invert pair would be as follows:

$$(I, (A - \sigma_1 B)^{-1}(B - \sigma_2 A)).$$

The most common choice is $\sigma_2 = 0$ and σ_1 which is close to an eigenvalue of the original matrix Saad (1992).

2.2. The Shift- and- Invert Arnoldi Method

If the matrix $A-\sigma B$ is invertible for some shift σ, the eigenproblem (*) can be transformed into the standard eigenproblem

$$A\phi_i = \lambda_i B\phi_i . \tag{1}$$

$$A\phi_i - \sigma B\phi_i = \lambda_i B\phi_i - \sigma B\phi_i .$$

$$(A-\sigma B)\phi_i = (\lambda_i - \sigma)B\phi_i \Rightarrow \frac{1}{\lambda_i - \sigma}\phi_i = (A-\sigma B)^{-1}B\phi_i .$$

Hence,

$$C\phi_i = \theta_i\phi_i , \tag{2}$$

where $\theta_i = \dfrac{1}{\lambda_i - \sigma}$.

It is easy to verify that (λ_i, φ_i) is an eigenpair of problem (*) if and only if (θ_i, φ_i) is an eigenpair of the matrix C. Therefore, the shift- and- invert Arnoldi method for the eigenproblem (1) is mathematically equivalent to the standard Arnoldi method for the transformed eigenproblem (2). It starts with a given unit length vector v_1 (usually chosen randomly) and builds up an orthonormal basis V_m for the krylov subspace $k_m(c, v_1)$ by means of the Gram-Schmidt orthogonalization process.

In finite precision, reorthogonalization is performed whenever same sever cancellation occurs Jia and Zhang (2002), Saad (1988). Then the approximate eigenpairs for the transformed eigenproblem (2) can be extracted from $k_m(c, v_1)$. The approximate solutions for problem (1) can be recovered from these approximate eigenpairs. The shift- and- invert Arnoldi process can be written in matrix form

$$(A-\sigma B)^{-1}BV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^* , \tag{3}$$

and

$$(A-\sigma B)^{-1}BV_m = V_m \tilde{H}_m , \tag{4}$$

where e_m is the m^{th} coordinate vector of dimension m, $V_{m+1} = (V_m, v_{m+1}) = (v_1, v_2, \ldots, v_{m+1})$ is an $n \times (m+1)$ matrix whose columns form an orthonormal basis of the $(m+1)$ dimensional krylov subspace $k_{m+1}(c, v_1)$, and $\widetilde{H}_m$ is the $(m+1) \times m$ upper Hessenberg matrix that is same as H_m expect for an additional row in which the only nonzero entry is $h_{m+1,m}$ in the position ($m+1, m$).

AAM: Intern. J., Vol. 5, Issue 1 (June 2010) [Previously, Vol. 5, No. 1]

Suppose that

$$(\tilde{\theta}_i, \tilde{y}_i), i = 1,2,\ldots,m$$

are the eigenpairs of the matrix H_m. Then,

$$H_m \tilde{y}_i = \tilde{\theta}_i \tilde{y}_i, \tag{5}$$

and

$$\tilde{\lambda}_i = \sigma + \frac{1}{\tilde{\theta}_i} \text{ and } \tilde{\phi}_i = V_m \tilde{y}_i. \tag{6}$$

When the shift-and-invert Arnoldi method uses $(\tilde{\lambda}_i, \tilde{\varphi}_i)$ to approximate the eigenpairs (λ_i, φ_i) of the problem (1), the $\tilde{\lambda}_i$ and $\tilde{\varphi}_i$ are called the Ritz values and the Ritz vectors of A with respect to $k_m(c, v_1)$. For details, refer to Ericsson and Ruhe (1980).

Defining the corresponding residual

$$\tilde{r}_i = (A - \tilde{\lambda}_i B)\tilde{\phi}_i. \tag{7}$$

Then, we have the following theorem:

Theorem 2.1:

The residuals $\tilde{r}_i$ corresponding to the approximate eigenpairs $(\tilde{\lambda}_i, \tilde{\varphi}_i)$ by the shift-and-invert Arnoldi method satisfy:

$$\|\tilde{r}_i\| \leq h_{m+1,m} \left|\tilde{\lambda}_i - \sigma\right| \|A - \sigma B\| \left|e_m^* \tilde{y}_i\right| \tag{8}$$

Proof:

From relations (3), (4) and (6), we obtain

$$\begin{aligned}
\|\tilde{r}_i\| &= \|(A-\tilde{\lambda}_i B)\tilde{\phi}_i\| = \|(A-\tilde{\lambda}_i B)V_m\tilde{y}_i\| = \|((A-\sigma B)-(\tilde{\lambda}_i-\sigma)B)V_m\tilde{y}_i\| \\
&= \|(A-\sigma B)(I-(\tilde{\lambda}_i-\sigma)(A-\sigma B)^{-1}B)V_m\tilde{y}_i\| \\
&= |\tilde{\lambda}_i-\sigma|\,\|(A-\sigma B)((A-\sigma B)^{-1}B-\tilde{\theta}_i I)V_m\tilde{y}_i\| \\
&\le |\tilde{\lambda}_i-\sigma|\,\|A-\sigma B\|\,\|V_{m+1}(\tilde{H}_m-\tilde{\theta}_i\tilde{I}_i)\tilde{y}_i\| \\
&= h_{m+1,m}\,|\tilde{\lambda}_i-\sigma|\,\|A-\sigma B\|\,|e_m^*\tilde{y}_i|.
\end{aligned}$$

3. A Technique for Computing Eigenvalues ($A\varphi_i = \lambda_i B\varphi_i$)

Below, we try to show the inverse matrix by LDV decomposition and block matrix. Then by selecting a suitable shift we try to find eigenvalues for $A\varphi_i = \lambda B\varphi_i$ in special interval [*Lb,ub*].

In the last section, we described that the shift-and-invert Arnoldi method for the eigenproblem $A\varphi_i = \lambda_i B\varphi_i$ is mathematically equivalent to the standard Arnoldi method for the transformed eigenproblem

$$(A-\sigma B)\varphi_i = (\lambda_i-\sigma)B\varphi_i \Rightarrow \frac{1}{\lambda_i-\sigma}\varphi_i = (A-\sigma B)^{-1}B\varphi_i \Rightarrow C\varphi_i = \theta_i\varphi_i\,,$$

or

$$A\phi_i = \lambda_i B\phi_i \rightarrow (A-\sigma B)^{-1}B\phi_i = \theta_i\phi_i, \qquad \theta_i = \frac{1}{\lambda_i-\sigma},$$

where σ is a shift. For computing $(A-\sigma B)^{-1}$, we can use block matrix method as follows:

In this method the matrix divided in $2-block \times 2-block$ matrix and by applying LDV decomposition, Datta (1994) the inverse is computed

$$M = (A-\sigma B) \qquad\qquad M = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ C_1A_1^{-1} & I \end{bmatrix}\begin{bmatrix} A_1 & 0 \\ 0 & S_1 \end{bmatrix}\begin{bmatrix} I & A_1^{-1}B_1 \\ 0 & I \end{bmatrix}$$

$$S_1 = (D_1 - C_1A_1^{-1}B_1)$$

$$M^{-1} = \begin{bmatrix} I & -A_1^{-1}B \\ 0 & I \end{bmatrix}\begin{bmatrix} A_1^{-1} & 0 \\ 0 & S_1^{-1} \end{bmatrix}\begin{bmatrix} I & 0 \\ -C_1A_1^{-1} & I \end{bmatrix} = \begin{bmatrix} A_1^{-1}+A_1^{-1}B_1S_1^{-1}C_1A_1^{-1} & -A_1^{-1}B_1S_1^{-1} \\ -S_1^{-1}C_1A_1^{-1} & S_1^{-1} \end{bmatrix}.$$

We can show that if the inverse of M exists, then the matrices A_1 and S_1 are invertible. In fact, by this decomposition instead of finding M^{-1} we compute the inverse of *L*, *D*, and *V*, which is much easier and faster than finding the inverse of M directly. In Matlab "inv" function (it calculates inverse matrix) requires $2n^3$ operations for a matrix with dimension n but we can see block inverse needs only n^3 operations. When we can find M^{-1} set $C = M^{-1}B$ and Arnoldi Algorithm can be used for solving $C\varphi_i = \theta_i \varphi_i$.

We choose $\sigma_1 = \dfrac{Lb+Ub}{2}$. If $M = (A - \sigma_1 B)$ is not invertible we set $Ub = \sigma_1$ and $\sigma_2 = \dfrac{Lb+Ub}{2}$. So, we bisect the interval [*Lb, ub*] to find a suitable shift.

Algorithm (Block Shift-and-Invert [BSI]):

Step 1: Input A, B, Lb, Ub;

Step 2: While $(Lb \neq Ub)$ do

(a) $\sigma = \dfrac{Lb+Ub}{2}$, $M = A - \sigma B$;

(b) **If** M is singular

$Ub = \sigma$;

go to (a) ;

Else go to the next step;

Step 3: Use Block Inverse method for computing M^{-1},

$$M = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ C_1 A_1^{-1} & I \end{bmatrix} \begin{bmatrix} A_1 & 0 \\ 0 & S_1 \end{bmatrix} \begin{bmatrix} I & A_1^{-1}B_1 \\ 0 & I \end{bmatrix},$$

$$S_1 = (D_1 - C_1 A_1^{-1} B_1),$$

$$M^{-1} = \begin{bmatrix} I & -A_1^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A_1^{-1} & 0 \\ 0 & S_1^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -C_1 A_1^{-1} & I \end{bmatrix} = \begin{bmatrix} A_1^{-1} + A_1^{-1}B_1 S_1^{-1} C_1 A_1^{-1} & -A_1^{-1}B_1 S_1^{-1} \\ -S_1^{-1} C_1 A_1^{-1} & S_1^{-1} \end{bmatrix};$$

Step 4: $C = M^{-1} * B$;

Step 5: Gain eigenvectors and eigenvalues (V, lm) of matrix C by Arnoldi Algorithm;

Step 6: **For** $i = 1$ to (rank matrix)

$$\ln(i) = \sigma + \frac{1}{lm(i)};$$

Step 7: **For** i=1 to (rank matrix)

If ($Lb \le \ln(i) \le Ub$)
$\ln(i)$ is the eigenvalue.
Else
("There are not any eigenvalues for input argument");

Step 8: Stop.

4. Sptarn

In this function the Arnoldi algorithm with spectral transformation is used.

$$[xv, lmb, iresult] = sptarn(A, B, Lb, Ub) .$$

This command finds eigenvalues of the equation $(A - \lambda B)x = 0$ in the interval $[Lb, Ub]$. A, B are $n \times n$ matrices, Lb and Ub are lower and upper bounds for eigenvalues to be sought. A narrower interval makes the algorithm faster. In the complex case, the real parts of lmb are compared to Lb and Ub. xv are eigenvectors, ordered so that norm ($A \times xv - B \times xv \times diag(lmb)$) is small. lmb is the sorted eigenvalues.

If $iresult \ge 0$ the algorithm succeeded and all eigenvalues in the intervals have been found . If $iresult < 0$ the algorithm is not successful, there may be more eigenvalues, try with a smaller interval. Normally the algorithm stops earlier when enough eigenvalues have converged. The shift is chosen at a random point in the interval [Lb, Ub] when both bounds are finite. The number of steps in the Arnoldi run depends on how many eigenvalues there are in the interval. After a stop, the algorithm restarts to find more Schur vectors in orthogonal complement to all those already found. When no eigenvalues are found in $Lb \le lmb \le Ub$, the algorithm stops. If it fails again check whether the pencil may be singular.

5. Numerical Tests and Comparisons

Sptarn Algorithm and Block Shift-and-Invert (BSI) Algorithm are tested for various matrices by Matlab Software. All tests are performed on a Intel(R) Celeron(R) M, CPU 1.46 GHZ Laptop, Matlab Version 7.5. We save eigenvalues in box [Lb, Ub] for different Matrix with different conditions. Sptarn Algorithm and BSI Algorithm are marked with (1) and (2), for example ir_1 shows the value "iresult" in Sptarn function and ir_2 shows the number of eigenvalues by BSI Algorithm.

- ir_1, ir_2, denote the number of eigenvalues in interval $[Lb, Ub]$
- t_1, t_2 are the CPU times in seconds
- F_1, F_2 are the smallest eigenvalue in $[Lb, Ub]$
- E_1, E_2 are the largest eigenvalue in $[Lb, Ub]$
- "n.c" failure to compute all the desired eigenvalues
- r_1, r_2 are residual ($norm(Ax - \lambda Bx)$) for the smallest eigenvalue

Example 1:

We were interested in finding the eigenvalues of $Ax = \lambda Bx$. A, B are sprandom, and unsymmetric matrices of different rank. Set a region $[Lb, Ub]$ with $Lb = -5 + i$, and $Ub = 5 + i$.

Table 1: Results of Example 1

Dimension	ir_1	ir_2	$t_1(s)$	$t_2(s)$	r_1	r_2	F_1	F_2	E_1	E_2
10x10	9	9	0.0514	0.0022	8.7155×10^{-15}	4.739×10^{-15}	-0.1347+0i	-0.1347+0i	-2.750700	-2.75070
20x20	19	19	0.0493	0.0053	9.7515×10^{-15}	8.6045×10^{-15}	0.0262+0i	0.0262+0i	3.4984+0i	3.4984+0i
50x50	50	50	0.1198	0.0673	9.6086×10^{-14}	3.6083×10^{-14}	-0.1509+0.0758i	-0.1509+0.0758i	-4.6028+0i	-4.6028+0i
100x100	98	98	0.5520	0.4170	2.8376×10^{-13}	2.1188×10^{-13}	-0.029+0i	-0.029+0i	4.1323-4.8307i	4.1323-4.8307i
300x300	-68	290	118.8775	10.3623	n.c	1.1161×10^{-12}	-0.8611+0.647i	0.0127+0.0435i	n.c	0.7588+9.8360i
500x500	-82	486	404.9588	49.5601	n.c	8.8866×10^{-12}	-0.224+0.3783i	0.0016+0i	n.c	-2.6521-7.3638i
1000x1000	-80	977	2.7393×10^{3}	395.7286	n.c	9.6210×10^{-11}	-0.1023+0.6254i	0.015+0i	n.c	-3.3883-16.8165i

It is seen from Table 1 that BSI Algorithm is much more efficient than Sptarn Algorithm in all cases. For example Sptarn Algorithm fails for some matrices such as n=300, 500, 1000.

Example 2:

In this example we assume A to be an ill conditioned matrix such as Hilbert and B is Identity matrix on region [0, 1].

Table 2: Results of Example 2

Dimension	Cond A	ir_1	ir_2	$t_1(s)$	$t_2(s)$	r_1	r_2	F_1	F_2	E_1	E_2
10x10	1.0625×10^{13}	9	9	0.5635	0.3804	3.6456×10^{-16}	3.3112×10^{-16}	1.0930×10^{-13}	1.0925×10^{-13}	0.3429	0.3429
20x20	2.9373×10^{18}	19	19	0.1922	0.0020	1.3004×10^{-14}	5.1988×10^{-16}	3.3307×10^{-16}	3.3307×10^{-16}	0.4870	0.4870
50x50	2.6060×10^{19}	20	39	0.1684	0.0070	7.375×10^{-16}	3.1470×10^{-16}	5.5511×10^{-17}	5.5511×10^{-17}	0.6797	0.6797
100x100	4.2276×10^{19}	26	74	0.2485	0.0694	8.9034×10^{-16}	3.735×10^{-16}	2.70756×10^{-17}	2.70756×10^{-17}	0.8214	0.8214
200x200	1.5712×10^{20}	20	125	0.1374	0.2500	1.4683×10^{-15}	3.5940×10^{-16}	1.2934×10^{-14}	1.2934×10^{-14}	0.9571	0.9571
500x500	5.1045×10^{20}	22	286	0.6536	4.2190	5.0413×10^{-15}	3.7338×10^{-16}	2.5535×10^{-15}	2.5535×10^{-15}	0.4056	0.4056
1000x1000	9.1197×10^{20}	25	590	2.9654	38.0649	5.9611×10^{-15}	1.5014×10^{-15}	1.1102×10^{-16}	1.1102×10^{-16}	0.4925	0.4925
2000x2000	3.7286×10^{21}	26	1192	17.0001	365.7796	8.3289×10^{-15}	8.2702×10^{-16}	3.1697×10^{-14}	3.1697×10^{-1}	0.5809	0.5809

We can see that BSI Algorithm works better than Sptarn Algorithm for large and ill conditioned matrices. For example we can compare columns ir_1 and ir_2 for large matrices. Numbers of eigenvalues that are gained by BSI Algorithm are more than the eigenvalues that are gained of Sptarn Algorithm in region [0, 1], for matrix with dimension 2000 Sptarn Algorithm finds only 26 eigenvalues in region [0, 1] but BSI Algorithm finds 1192 eigenvalues in this interval.

Example 3:

Has been taken from Bai and Barret (1998), Consider the constant coefficient convention diffusion differential equation

$$-\Delta u(x,y)+p_1u_x(x,y)+p_2u_y(x,y)-p_3u(x,y)=\lambda u(x,y)$$

On a square region $[0,1]\times[0,1]$. With the boundary condition $u(x,y)=0$, where p_1, p_2 and p_3 are positive constants discretization by five point finite differences on uniform $n\times n$ grid points using the row wise natural ordering gives a block tridiagonal matrix of the form

$$A=\begin{bmatrix} T & (\beta+1)I & & & \\ (-\beta+1)I & T & (\beta+1)I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & (\beta+1)I \\ & & & (-\beta+1)I & T \end{bmatrix}$$

with

$$T = \begin{bmatrix} 4-\tau & \gamma-1 & & & \\ -\gamma-1 & 4-\tau & \gamma-1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma-1 \\ & & & -\gamma-1 & 4-\tau \end{bmatrix},$$

where $\beta = (1/2)p_1 h,\ \gamma = (1/2)p_2 h,\ \tau = p_3 h^2$ and $h = 1/(n+1)$. The order of A is $N = n^2$. By taking $p_1 = 1$, $p_2 = p_3 = 0$ and B = Identity matrix, for different order on the region [5,7] we have Table 3.

Table 3. Results of Example 3

Dimension	ir_1	ir_2	$t_1(s)$	$t_2(s)$	r_1	r_2	F_1	F_2	E_1	E_2
9x9	3	3	0.5072	0.1179	$5.9618\text{x}10^{-16}$	$4.4409\text{x}10^{-16}$	5.4142	5.4142	5.4142	5.4142
36x36	2	12	0.0619	0.0043	$2.2767\text{x}10^{-14}$	$1.6398\text{x}10^{-15}$	5.247	5.2470	5.8019	50.8019
100x100	30	30	0.2147	0.0204	$2.2828\text{x}10^{-15}$	$2.0940\text{x}10^{-15}$	5.3097	5.3097	5.9190	5.9190
225x225	72	75	1.5919	0.2692	$5.0565\text{x}10^{-15}$	$2.5288\text{x}10^{-15}$	5.1111	5.1111	5.9616	5.9616
400x400	95	139	9.6975	1.5786	$8.5051\text{x}10^{-15}$	$3.9222\text{x}10^{-15}$	5.4661	5.0000	5.9777	5.9777
900x900	99	300	37.7165	18.9227	$4.6226\text{x}10^{-15}$	$5.5997\text{x}10^{-15}$	5.6415	5.0579	5.9897	5.9897
1600x1600	61	520	157.3872	116.0561	$2.6665\text{x}10^{-15}$	$9.0427\text{x}10^{-15}$	5.0871	5.0871	5.3307	5.9941
2500x2500	73	834	369.7465	551.2750	$3.9148\text{x}10^{-15}$	$1.5793\text{x}10^{-14}$	5.1047	5.0000	5.2053	5.9962

As we can see BSI Algorithm gives more eigenvalues in less time and with higher accuracy than Sptarn Algorithm for different matrices.

Example 4:

Has been taken from Bai and Barret (1998) Dielectric channel waveguide problems arise in many integrated circuit applications.

Discretization of the governing Helmholtz equation for the magnetic field H

$$\nabla^2 H_x + k^2 n^2(x,y) H_x = \beta^2 H_x$$
$$\nabla^2 H_y + k^2 n^2(x,y) H_y = \beta^2 H_y.$$

By finite difference leads to an unsymmetric matrix eigenvalue problem of the form

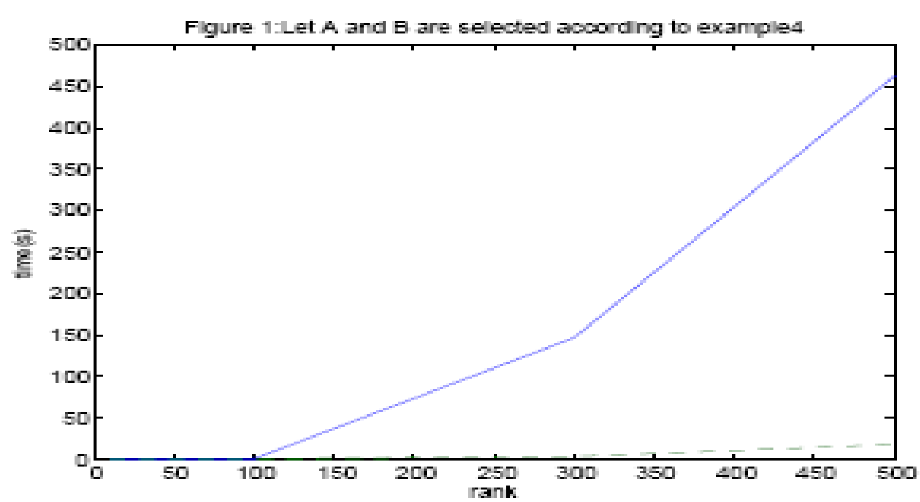
$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}\begin{bmatrix} H_x \\ H_y \end{bmatrix} = \beta^2 \begin{bmatrix} B_{11} & \\ & B_{22} \end{bmatrix}\begin{bmatrix} H_x \\ H_y \end{bmatrix},$$

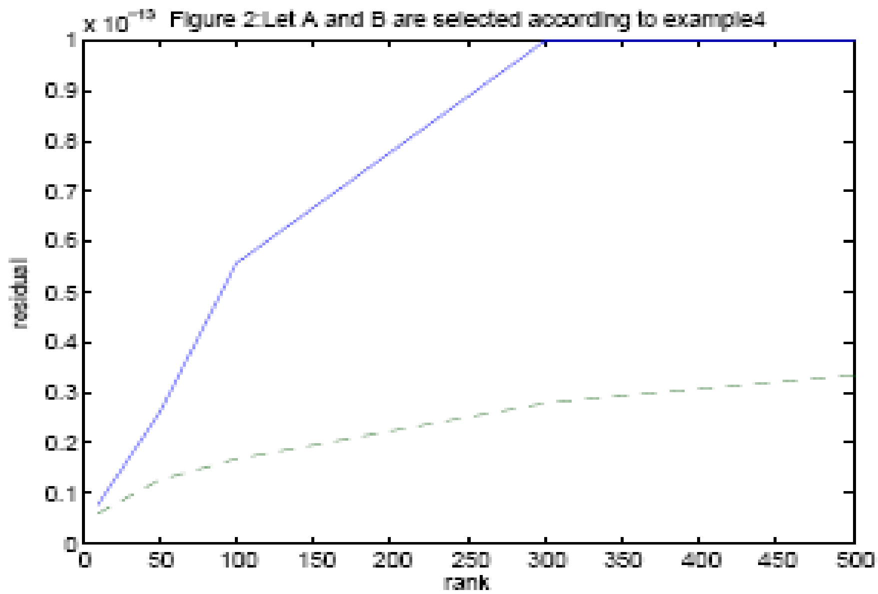
where C_{11} and C_{22} are five- or- tridiagonal matrices, C_{12} and C_{21} are (tri-) diagonal matrices, B_{11} and B_{22} are nonsingular diagonal matrices. The problem has been tested in the region [0, 10].

Table 4: Results of Example 4

Dimension	ir_1	ir_2	$t_1(s)$	$t_2(s)$	r_1	r_2	F_1	F_2	E_1	E_2
10x10	10	10	0.0504	0.0018	7.7702×10^{-15}	6.0311×10^{-15}	1.3692-(0.0871)i	1.36920-(0.0871)i	8.3760	8.3760
50x50	50	50	0.0678	0.0181	2.6101×10^{-14}	1.2672×10^{-14}	1.0145-(0.1702)i	1.0145-(0.1702)i	8.9395	8.9395
100x100	100	100	0.2743	0.1334	5.575×10^{-14}	1.6864×10^{-14}	1.0038-0.087i	1.0038-0.087i	8.9634	8.9634
300x300	n.c	300	147.1122	3.8444	n.c	2.7922×10^{-14}	n.c	1.0004-0.0294i	n.c	8.9708
500x500	n.c	500	463.1583	18.5093	n.c	3.3542×10^{-14}	n.c	1.0002-0.0177i	n.c	8.9714

We can see Sptarn Algorithm failed to compute the desired eigenvalues for some matrices. The results of this example are plotted as Figure1 and Figure2. The broken lines are shown the results of BSI Algorithm and connected lines are shown the results of Sptarn Algorithm.





6. Conclusions

In this paper, we have considered Block Shift-and-Invert (BSI) Algorithm. In this method we compute $M^{-1} = (A - \sigma B)^{-1}$. This computation has been done by block matrix and a suitable shift. As we have shown that if we need all eigenvalues in interval [*Lb, ub*] or close to a given shift for singular matrix, the BSI Algorithm obtains them with very high speed and accuracy. All numerical examples have been compared with corresponding results from Sptarn Algorithm in Matlab. It has been shown that BSI results were much more efficient than Sptarn results.

Acknowledgments

The authors would like to thank the referees for their comments which improved the presentation of the paper.

REFERENCES

Bai, Z., Barret, R., Day, D., Demmel, J., and Dongarra (1998). Test matrix collection for non-Hermitian Eigenvalue problems matrix market.

Datta, B. N. (1994). Numerical Linear Algebra and Applications, ITP, an International Thomson Company.

Ericsson, T. and Ruhe, A. (1980). The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems, Math. Com., Vol. 35, pp. 1251- 1268.

Golub, G. H. and Van Loan, C. F. (1989). Matrix Computations, The John Hopkins University press USA.

Jia, Z. and Zhang, Y. (2002). A Refined shift- and- invert Arnoldi Algorithm for large unsymmetric Generalized Eigen problems, Computers and Mathematics with Applications, Vol. 44, pp. 1117- 1127.

Ruhe, A. (1994). Rational Krylov algorithms for nonsymmetric eigenvalue problems.ii Matrix pairs. Lin. Alg. and its Appl., Vol. 197, pp. 283-295.

Saberi Najafi, H. and Shams Solary, M. (2005). Shifting algorithms with maple and implicit shift in the QR algorithm, Applied Mathematics and Computation, Vol. 161, pp. 947-962

Saad, Y. (1988). Variations on Arnoldi's method for computing eigen elements of large unsymmetric matrices, Linear Algebra and its Applications, Vol. 34, pp. 269-295.

Saad, Y. (1992). Numerical methods for large eigenvalue problems, Theory and Algorithms, Wiley, New York.