ON WAYS TO IMPROVE CONVERGENCE OF KRYLOV SUBSPACE METHODS

by

SALIM ABDULLAH SURUR AL-KHARUSI

Dissertation submitted in partial fulfillment
of the requirements for the degree
of Master of Science in Mathematics

MAY 2008

BERKENAAN CARA-CARA MEMPERBAIKI PENUMPUAN KAEDAH SUBRUANG KRYLOV

ABSTRAK

"SNAP" atau "Solution by Null-space Approximation and Projection" ialah salah satu cara untuk menyelesaikan sistem linear apabila pekali matrik adalah besar dan "sparse". Objektifnya ialah untuk mengatasi masalah penumpuan yang perlahan atau genangan yang disebabkan oleh nilai eigen yang kecil.

Dissertasi ini bertujuan untuk menyediakan satu sorotan kritikal tentang kaedah "SNAP" yang dicadangkan pada 2006 oleh M. Illic, W. Turner dan Y. Saad. Dissertasi ini memfokuskan pada idea utama "SNAP" : algoritmanya, pembinaan Penghampiran Ruang nol hampiran dan dua algoritma yang dihasilkan oleh Illic. W. Turner dan Y. Saad iaitu SNAP-JD(m) dan 'Restarted SNAP-JD($m, k_{\max}, l$)

Kajian ini juga menyediakan sorotan literature tentang kajian lain yang cuba untuk mengatasi masalah penumpuan perlahan atau genangan dengan memfokuskan pada nilai eigen yang kecil ketika menyelesaikan sistem linear apabila pekali matrik besar dan sparse

ABSTRACT

"SNAP" or "Solution by Null-space Approximation and Projection" is one of the methods for solving linear system when the matrix coefficient is large and sparse. Its objective is to overcome the problem of slow convergence or stagnation which is caused by small eigenvalues.

This dissertation is aimed at providing a critical review of the SNAP method which was proposed in 2006 by M. Illic, W. Turner and Y. Saad. The dissertation focused on the main idea of SNAP: the algorithm, the construction of Approximate Null Space and two algorithms generated by Illic, W. Turner and Y. Saad which are SNAP-JD(m) and Restarted SNAP-JD($m, k_{\max}, l$).

This study also provide literature reviews of other works that try to overcome the problem of slow convergence or stagnation by focusing in small eigenvalues while solving a linear system when the matrix coefficients is large and sparse.

CHAPTER 1

INTRODUCTION

Finding the solution to the system

$$Ax = b \tag{1.1}$$

when the $n \times n$ coefficient matrix A is large and sparse, is one of the main problem in Numerical Linear Algebra. Mathematicians try to find different iterative methods which approximate the solution of the system (1.1). They always keep in mind two important things, i) rate of convergence to get the approximate solution ii) how good the approximation is compared to exact solution?.

In November 2006 Ilic, Turner and Saad published a new method to solve the system (1.1) when A is large and sparse. The new method is called "Solution by Null-Space Approximation and Projection". Since the convergence of the iteration method is sometimes hampered by the presence of small eigenvalues they –in the new method- focus the attention directly on the "small" eigenspace (singular vector space).

One of the objectives of this dissertation review some methods that try to overcome the problem of slow convergence or stagnation that caused by presence of small eigenvalues. Another objective of this dissertation is to analyses the SNAP method which present a new idea to improve the convergence in Krylov algorithm (GMRES).

The most popular general-purpose of the iterative solution technique for solving a system of linear equation $Ax = b$, when the coefficient matrix A is large and sparse, is to utilize a combination of preconditioning with Krylov subspace methods. Krylov's methods such as Generalized Minimal Residual Method (GMRES) construct a subspace

$$K_m(A,b) = span\{b, Ab, A^2b, ..., A^{m-1}b\}$$

for which an approximation solution is extracted (Ilic et.al ,2006).

This dissertation is organized as follows. In section 2, a general definition for Krylov subspace is presented and steps to produce a Krylov subspace is described. An example of a Krylov method is also described. In section 3, literature review of research in the problem of small eigenvalues is presented. The main section in this dissertation is section 4, in which the analysis and the algorithm for SNAP are illustrated.

So before we start to study and analyze the SNAP method, first we introduce the Krylov subspace and we give one example of Krylov methods "GMRES".

CHAPTER 2

KRYLOV SUBSPACE

A popular method for solving a system of linear equations (1.1) when the $n \times n$ coefficient matrix A is large and spars, $x, b \in R^n$, is to use Krylov subspace $K_k(A,b) = \{b, Ab, A^2b, ..., A^{k-1}b\}$ (Ilic & Tuner,2004).

2.1. GENERAL DEFINITION

Let x_0 and $r_0 :\equiv b - Ax_0$ be an initial guess of the solution and the corresponding residual, respectively, most Krylov methods generate a sequence of approximations or iterates x_n with property that

$x_n - x_0 = K_k(A, r_0) :\equiv span\{r_0, Ar_0, ..., A^{k-1}r_0\}$.

Here $K_k(A, r_0)$ is the Krylov space generated by A from r_0 . There are also Krylov space method using different spaces, like $K_k(A^T, r_0)$ or $K_k(AA^T, r_0)$. (Gutknecht & Rozloznik , 2001)

Krylov subspace methods are iterative methods in which at the n^{th} step an approximation to the solution of (1.1), x_n is found in x_0+K_n. This approximation is found by requiring x_n to be the minimizer of some functional. Different methods depend on the choice of this functional, characteristic of the matrix , and on some implementation details (Simoncini & Szyld, 2005).

An iterative scheme for solving linear system (1.1) is called a Krylov subspace method if it produces approximate solutions of the form

$$(2.1) \qquad x_m = x_0 + K_m(A, r_0), \qquad m = 1, 2, \ldots,$$

Where $x_0$is an arbitrary initial guess with the corresponding residual vector $r_0 = b - Ax_0$ and $K_m(A, r_0)$ is the m^th^ Krylov subspace defined by $K_m(A, r_0) = span\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$ (Wei & Wu, 2000).

There are two main steps in designing a Krylov subspace method. The first step is the construction of suitable vectors $v_1, v_2, \ldots, v_m$ that span $K_m(A, r_0)$. Then setting $V_m = [v_1, v_2, \ldots, v_m]$, we parameterize the m$^{\text{th}}$ iteration (1.2) as follows:

$$(2.2) \qquad x_m = x_0 + V_m z_m, \quad \text{where} \qquad z_m \in R^m.$$

The second step, it remains to specify the choice of $z_m,$ in (2.2). Different strategies in the two steps above lead to different Krylov subspace methods. Some examples are generalized minimal residual algorithm (GMRES), biconjugate gradient algorithm (BICG),QMR and TFQMR, etc. An important feature of the Krylov methods is that they involve only multiplications of vector with matrices A and A* (conjugate transpose of A). (Wei & Wu, 2000)

How do you solve a system of linear equations $^{(1.1)}$ when your coefficient matrix A is large and sparse (i.e., contains many zero entries)? What if the order n of the matrix is so large that you cannot afford to spend about n^3 operations to solve the system by Gaussian elimination? Or what if you do not have direct access to the matrix? Perhaps the matrix A exists only implicitly as a subroutine that, when given a vector v , returns Av .

In this case you may want to use a Krylov method. Krylov methods are used in numerical as well as in symbolic computation . Since there is no universally agreed definition, we say here that a *Krylov method* solves $^{(1.1)}$ by repeatedly performing matrix-vector multiplications involving A (this excludes methods like Lanczos biorthogonalization, QMR, and biconjugate gradient methods that also require matrix-vector multiplications involving the conjugate transpose A*).

Starting with an initial guess x_0 , a Krylov method bootstraps its way up (we hope!) to ever more accurate approximations x_k to a desired solution. In iteration k a Krylov method produces an approximate solution x_k from a Krylov space generated by a vector c,

$$K_k(A,c) = \{c, Ac, A^2c, \ldots, A^{k-1}c\}$$

A popular choice is $c = b$ (because one can obtain convergence estimates, and because there is often no other problem-dependent guess) and $x_0 = 0$. That's why we restrict ourselves to Krylov spaces $K_k(A, b)$ that are generated by the right-hand side b of a linear system (1.1) (Ipsen & Meyer,).

2.2. AN EXAMPLE OF KRYLOV METHOD

General Minimal Residual (GMRES):

The General Minimal Residual (GMRES) is chosen as an example because the SNAP method was built on this method. "GMRES" method is the main part in SNAP algorithm. The general minimal residual method (GMRES) was published by Saad and Schultz in 1986 (Ipsen and Meyer).

The next theorem relates Hessenberg matrix H_m that entries are defined by Arnoldi algorithm and Q_m which is also generated by Arnoldi algorithm. We will use this relation to deduce relation (3.4).

THEOREM 2.1: (Trefethen & Bau, 1997)

The matrices Q_m generated by the Arnoldi iteration are reduced QR factors of the Krylov matrix $K_m = \left[b \middle| Ab \middle| \ldots \middle| A^{m-1}b\right]$:

$$K_m = Q_m R_m$$

The Hessenberg matrices H_m are the corresponding projections

$$H_m = Q_m^* A Q_m$$

and the successive iterates are related by the formula

$$A Q_m = Q_{m+1} \overline{H}_m .$$

The general minimal residual method (GMRES) is a projection method based on taking $\mathcal{K} = \mathcal{K}_m$ and $\mathrm{L} = A\mathcal{K}_m$ in which $\mathcal{K}_m$ is the m$^{\mathrm{th}}$ Krylov subspace with $\upsilon_1 = r_0 / \|r_0\|_2$ .Such technique minimize the residual norm over all vectors in $x_0 + \mathcal{K}_m$.

Any vector x in $x_0 + \mathcal{K}_m$ can be written as

(2.3) $x = x_0 + V_m y$

Where y is an m-vector. Defining

(2.4) $J(y) = \|b - Ax\|_2 = \|b - A(x_0 + V_m y)\|_2$

By using the relation $AV_m = V_{m+1}\overline{H}_m$ where $\overline{H}_m$ is the (m+1) × m Hessenberg matrix whose nonzero entries h_{ij} are defined by Arnoldi algorithm.

$$b - Ax = b - A(x_0 + V_m y)$$

$$= r_0 - AV_m y$$

$$= \beta \upsilon_1 - V_{m+1}\overline{H}_m y$$

(2.5) $$=V_{m+1}(\beta e_1-\overline{H}_m y).$$

Since the column vectors of V_{m+1} are orthonormal, then

(2.6) $$J(y)=\|b-A(x_0+V_m y)\|_2=\left\|\beta e_1-\overline{H}_m y\right\|_2.$$

The GMRES approximation is the unique vector of x_0+K_m which minimize (2.4). By (2.3) and (2.6), this approximation can be obtained quite simply as $x_m=x_0+V_m y_m$ where y_m minimizes the function $J(y)=\left\|\beta e_1-\overline{H}_m y\right\|_2$ i.e.,

$$x_m=x_0+V_m y_m \quad \text{where} \quad y_m=\arg\min_y \left\|\beta e_1-\overline{H}_m y\right\|_2$$

The Algorithm of GMRES:(Saad,2000)

```
1. Compute $r_0=b-Ax_0, \beta:=\|r_0\|_2$ and $\upsilon_1=r_0/\beta$

2. Define the (m+1) × m matrix $\overline{H}_m=\{h_{ij}\}_{1\le i\le m+1, 1\le j\le m}$. Set $\overline{H}_m=0$.

3. For j= 1,2,3,…,m Do:

4.     Compute $w_j:=A\upsilon_j$

5.     For i= 1,…,j Do:

6.         $h_{ij}=(w_j,\upsilon_j)$
```

```
7.          $w_j = w_j - h_{ij}\upsilon_j$
8.    EndDo
9.    $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ set m:= j and go to 12
10.   $\upsilon_{j+1} = w_j / h_{j+1,j}$
11. EndDo
12. Compute $y_m$ that minimizes $\|\beta e_1 - \overline{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$.
```

CHAPTER 3

LITERATURE REVIEW

This section describes two main methods that try to overcome the problem of slow of convergence and both methods focusing on small eigenvalues.

3.1 Restarted General Minimal Residual preconditioned by deflation (Erhel et.al , 1996) :

The restarted GMRES preconditioned by deflation is one of the methods which try to overcome the problem of slow convergence in Krylov algorithm which caused by small eigenvalues. The convergence behavior of full-GMRES version analyzed and its found super linear convergence ,but, because of memory requirement a restarted version must be used in general (Erhel et.al , 1996). The full-GMRES version behaves as if the smallest eigenvalues are removed after some iteration(Erhel et.al , 1996). But this is no longer true in the restarted case. Therefore, the aim of "Restarted GMRES Preconditioned by Deflation" is to remove them by preconditioner. After each cycle of GMRES, the preconditioner is updated by pulling out new eigenvalues.

In this method (Restarted GMRES Preconditioned by Deflation) the authors built the "Preconditioner" M. The linear system (1.1) preconditioned from the right (in this method) they used a right preconditioning in order to guarantee a non-increasing residual. The objective is to remove the smallest eigenvalues of A which are known to slow down the convergence of GMRES (Erhel et.al , 1996) and replace them by real positive eigenvalues equal to the largest modulus of the eigenvalues. The preconditioning matrix will have multiple eigenvalue equal to this largest modulus and the eigenvalues of the original matrix which are not removed.

Theorem(3.1) (Erhel et.al , 1996)

If $T = U^T A U$ and $M = I_n + U(1/|\lambda_n| T - I_r)U^T$, then M is nonsingular and $M^{-1} = I_n + U(|\lambda_n| T^{-1} - I_r)U^T$ and the eigenvalues of AM^{-1} are $\lambda_{r+1}, \lambda_{r+2}, \dots, \lambda_n, |\lambda_n|, |\lambda_n|, |\lambda_n|$ with a multiplicity at least r.

So the matrix M which build in this way guarantee that the eigenvalues of the preconditioned matrix AM^{-1} will be in the form $\lambda_{r+1}, \lambda_{r+2}, \dots, \lambda_n, |\lambda_n|, |\lambda_n|, |\lambda_n|$ where $\lambda_{r+1}, \lambda_{r+2}, \dots, \lambda_n$ the eigenvalues of the original matrix A which are not removed. The matrix M is used in new Restarted GMRES , right preconditioned by deflating called DEFLGRES(m, l).

Example1: (Moriya and Nodera, 2000)

Consider the following linear system of equation $Ax = b$:

$$A = \begin{pmatrix} 1 & 0.1 & & & 0 \\ & 2 & 0.1 & & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & & 0.1 \\ 0 & & & & 16384 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}, \text{ where } A \in R^{16384\times16384}$$

The system was solved using standard GMRES(m) and DEFLATED-GMRES(m, l). The test uses two subspaces dimension of 50 and 70, the last 2 and 4 of which are approximate eigenvalues. The convergence results of this algorithm are given in the next table.

TABLE1: The numerical result in Example 1:

Algorithm	Time (sec)	Number of iterations
GMRES(50)	418	4100
GMRES(70)	450	3220
DEFLATED-GMRES(50,2)	358	3250
DEFLATED-GMRES(50,4)	261	2250
DEFLATED-GMRES(70,2)	330	2240
DEFLATED-GMRES(70,4)	260	1680

We can deduce following from the table:

i) GMRES(m) with largest m need less number of iterations to capture the approximate solution than small m.

ii) DEFLATED-GMRES(m,l) needs less time and less number of iterations to capture the approximate solution than GMRES(m) for same m.

iii) DEFLATED-GMRES(70,4) performs better to capture the approximate solution (need less time and less number of iterations) than DEFLATED-GMRES(70,2).

3.2 Jacobi-Davidson Iteration Methods (Sleijpen & Vorst, 2000)

The Jacobi-Davidson Iteration method is another method that has improved convergence properties and that may be used for general matrices (symmetric and non symmetric matrices). This new method is essentially a combination of the Jacobi Orthogonal Component Correction JOCC method and Davidson method.

3.2.1 Davidson Method:

Suppose we have some subspace K of dimension k, over which the projected matrix A has a Ritz value θ_k (e.g. θ_k is the largest Ritz value) and a corresponding Ritz vector u_k, (see definition A.6) and assume that an orthogonal basis of K is given by $\upsilon_1, \upsilon_2, \ldots, \upsilon_k$. The idea of the Davidson method is to expand the subspace in order to update u_k.

We can summarize the approach in the following steps:

1) Compute the residual $r_k = Au_k - \theta_k u_k$
2) Compute t from $(D_A - \theta_k I)t = r_k$, where D_A is a diagonal of the matrix A.
3) Make the vector t to be orthogonal to the basis vector $\upsilon_1, \upsilon_2, \ldots, \upsilon_k$ and the result is chosen.
4) Chose the new vector to be the new υ_{k+1} , by which K is expanded.

3.2.2 Jacobi Method:

Jacobi introduced a combination of two iterative methods for the computation of approximation of eigenvalues of a symmetric matrix. The first method is "Jacobi's Diagonalization Method". It is based on Jacobi plane rotation, which are used to force the matrix A to be diagonal dominant. The second method is JOCC and it is related to the original Davidson method with diagonal scaling.

The idea of "Jacobi Iteration" is that , the Jacobi iteration converges (fast) if the matrix is (strongly) diagonally dominant. Therefore Jacobi proposed to perform a number of steps of the Jacobi Diagonalization Method in order to obtain a (strongly) diagonal dominant matrix before applying the JOCC method.

Suppose we have a diagonal dominant matrix A, of which $a_{1,1} = \alpha$ is the largest diagonal element. Then α is an approximation for the largest eigenvalue λ , and e_1 is an approximation for the corresponding eigenvector u .

The application of diagonalization method can be viewed as a technique to improve the guess e_1 , i.e. the given matrix is rotated so that e_1 is closer to the (rotated) eigenvector u .

The (k+1)th step in either JOCC or Davidson method can be summarized as follows:

JOCC: Jacobi computed the component $\hat{y}_k$ of t_k orthogonal to u_1 and takes $u_{k+1} = u_k + \hat{y}_k$, $\theta_{k+1} = e_1^T A u_{k+1}$. Unlike Davidson's approach, Jacobi only computes components that are orthogonal to $u_1 = e_1$.

Davidson's Method: Davidson computes the component υ_{k+1} of t_k orthogonal to $u_1, u_2, \ldots, u_k$ and takes for u_{k+1} (and θ_{k+1}) the Ritz vector (respectively Ritz value) of A with respect to the space spanned by $\upsilon_1, \upsilon_2, \ldots, \upsilon_{k+1}$.

Davidson exploits the complete subspace constructed so far, while Jacobi only takes a simple linear combination of the last vector and the last correction. .

The resulting algorithm (the Jacobi-Davidson iteration method) maybe viewed as a combination of Jacobi approach to looking for the orthogonal complement of a given eigenvector approximation and Davidson algorithm for expanding the subspace in which the eigenvector approximation are constructed.

CHAPTER 4

SOLUTION BY NULL-SPACE APPROXIMATION AND PROJECTION (SNAP)

The basic idea of SNAP is to transform the non-homogeneous system $Ax = b$ to a homogeneous system $\overline{A}x = 0$. But why do we need to do that ? Ilic, Turner and Saad in 2006 claimed that "the eigenvalue problem for small eigenpairs has to be solved to improve the convergence rate of the Krylov subspace method for solving the linear system $Ax = b$."

So in their method embedding the solution of the linear system $Ax = b$ into an eigenvalues problem, or to be more accurate a null-space vector problem i.e. to find the solution of the system $\overline{A}x = 0$.

After transforming the system $Ax = b$ to $\overline{A}x = 0$ the solution x is a null space vector of $\overline{A}$. So this way of thinking enables one to consider powerful strategies utilized in solving eigenvalue problems like Generalized Minimal Residual or Jacobi-Davidson (JD). For this reason the Jacobi-Davidson(JD) method is used as fundamental method in SNAP algorithm. This gives theoretical motivation for SNAP for overcoming poor convergence (Ilic, Turner and Saad, 2006).

4.1. ALGORITHM OF SNAP

The algorithm is summarized below in three main steps.

Step1: Building a projector E, by which transforms $Ax = b$ into $\overline{A}x = 0$ where $\overline{A} = EA$ and $Eb = 0$, E is called the annihilator of b.

Step2: A sequence of "approximate null subspace " (ANSs) of dimension k and order ε is generated for $\overline{A}$.

$$N(\overline{A}, k, \varepsilon) = \{span\{w_1, w_2, \ldots, w_k\} \mid \|\overline{A}W\| < \varepsilon, W = [w_1, w_2, \ldots, w_k]\}$$

Step3: Extract the approximate solution which is a scalar multiple of the ANS vector.

4.2 Examples Of the Annihilators

The annihilator of b is a projector E such that $Eb = 0$. The annihilator for a given b is not unique. The next proposition is to derive the bound on the residual error $\|Ax - b\|$ in terms of the error $\|EAx\|$.

PROPOSITION 4.1: (Ilic et.al ,2006)

Define the orthogonal projector $E_{\perp} = I - bb^T / \|b\|^2$ and let a unit vector w be given which satisfies $\|E_{\perp}Aw\| = \sigma$. Assume that $b^T Aw \neq 0$ and let $\beta = \|b^2\| / (b^T Aw)$ and $x = \beta w$. Then the linear system $Ax = b$ has approximate solution x with residual $r = b - Ax$ satisfying $\|r\| = |\beta| \sigma$.

Proposition 4.1, is used in SNAP-JD in section 4.4 and in Restarted SNAP-JD in section 4.5 in the last step to compute the approximate solution x .

ORTHOGONAL PROJECTOR: A projector P is orthogonal if and only if it is self-adjoint ,which mean that, in the context of real vector spaces, the associated matrix is symmetric relative to orthonormal basis: (P^2=P). Indeed, if x is a vector in the domain of projection, then $Px \perp x - Px$ i.e. $\langle Px, x - Px\rangle = (Px)^T(x - Px) = x^T(P - P^2)x = \langle x, 0\rangle = 0$

(http://en.wikipedia.org/wiki/Orthogonal_projection#Orthogonal_projections)

EXAMPLE 1: Let $E_\perp = I - bb^T / \|b\|^2$ is an orthogonal projector and let $b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$.

Then $E_\perp = \begin{pmatrix} 9/13 & -6/13 \\ -6/13 & 4/13 \end{pmatrix}$ and $E_\perp b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. If we take $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ a vector in the domain of the projection, then $E_\perp x = \begin{pmatrix} 9/13 \\ -6/13 \end{pmatrix}$ and $x - E_\perp x = \begin{pmatrix} 4/13 \\ 6/13 \end{pmatrix}$, and we can see $E_\perp x \perp x - E_\perp x$.

OBLIQUE PROJECTOR: The term oblique projections is sometimes used to refer to non-orthogonal projections.

(http://en.wikipedia.org/wiki/Orthogonal_projection#Oblique_projections)

EXAMPLE 2: Let $E = I - (1/\|b\|_1)be^T$ where e is the vector with components

$$e_j = \begin{cases} 1 & \text{if } \mathrm{b}_j \geq 0 \\ -1 & \text{if } \mathrm{b}_j < 0 \end{cases}$$

and let $b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$. Then $e = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $E = \begin{pmatrix} 3/5 & -2/5 \\ -3/5 & 2/5 \end{pmatrix}$ and $Eb = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

If we take $x = \begin{pmatrix}1\\0\end{pmatrix}$ a vector in the domain of the projection, then $Ex = \begin{pmatrix}3/5\\-3/5\end{pmatrix}$ and $x - E_{\perp}x = \begin{pmatrix}2/5\\3/5\end{pmatrix}$, but Ex is not orthogonal to $x - Ex$.

EXAMPLE 3: $E_{|} = I - (1/b_j)b_j e_j^T$ where $|b_j| = \|b\|_\infty$ is other example of oblique projector.

4.3. CONSTRUCTION OF AN APPROXIMATE NULL SPACE

To obtain ANS vector of $\bar{A}x = 0$ we can exploit the GMRES algorithm. GMRES will not break down unless the solution is found.

The following lines give a more detailed description:

1) Select a unit vector υ_0 and set $\upsilon_1 = \bar{A}\upsilon_0$.
2) Process Arnoldi up to some specified point ℓ , resulting in the relation $\bar{A}V_\ell = V_{\ell+1}\bar{H}_\ell$, $V_\ell = [\upsilon_1, \upsilon_1, \dots, \upsilon_\ell]$
3) Compute the minimizer y_ℓ of $\min_{y \in R^\ell} \|\bar{A}V_\ell y - \upsilon_1\|$ as $y_\ell = \|\upsilon_1\| \bar{H}_\ell^{\dagger} e_1$ where $\bar{H}_\ell^{\dagger}$ is the pseudoinverse of $\bar{H}_\ell$.
4) The desired ANS vector is given by $w_1 = \upsilon_0 - V_\ell y_\ell$.

At this stage, examine whether $\|\bar{A}w_1\|$ is small and if that is the case, then check that $\|r\| = |\beta| \|\bar{A}w_1\| < \varepsilon$. If the residual is sufficiently small so $x = \beta w_1$ is an approximate solution of the linear system. If that is not true w_1 is a suitable candidate to be used as a vector in the subspace expansion Jacobi-Davidson-like method discussed later.

For initial approximation υ_0 the procedure above to produce an ANS vector w_1 may not be sufficiently accurate and iterative procedure has to be established to further refine w_1 to the desired null-space vector.

Two methods suggest possible ways for this refinement:

1) **Method I:**

Generate a sequence $\{w_m\}$ such that $\|\bar{A}w_m\| \to 0$, then $\{w_m\} \to z \in \mathcal{N}(\bar{A})$. The basic algorithm SNAP-JD with (k=k$_{\max}$ fixed) construct a sequence such that $\|\bar{A}w_{m+1}\| \leq \|\bar{A}w_m\|$, which is refered to as a **"Monotone Decreasing Sequence"**.

2) **Method II :**

Generate a sequence of ANS for $\bar{A}$:

$\{span\{w_1, w_2, \ldots, w_k\} \mid \|\bar{A}W\| < \varepsilon, W = [w_1, w_2, \ldots, w_k]\}$ where w_i's are orthonormal and ε is decreasing. The Restarted SNAP-JD in the next chapter constructs a sequence of subspace such that $\|\bar{A}W^{m+1}\| \leq \|\bar{A}W^m\|$ which is refered to as **"Monotone Decreasing Sequence of subspace"**.

Method I and method II form the basis for deriving SNAP-JD and Restarted SNAP-JD which will be described in sections 4.4 and 4.5 .

4.4. BASIC SNAP-JD

According to method I that mentioned in section(4.3) , given , w_i we need a procedure to construct w_{i+1} such that $\|\bar{A}w_{i+1}\| \leq \|\bar{A}w_i\|$. Corollary A.5 indicates that expanding a subspace by an orthonormal vector will result in an improved convergence.

The main idea in SNAP-JD algorithm is to use Jacobi-Davidson idea for choosing the expansion vector.

We can summarize the SNAP-JD algorithm in the following steps:

1- Start with initial vector w_i .

2- Construct a sequence of expanding subspace.

3- Obtain the corresponding SVD.

4- Test whether the smallest singular value is sufficiently small.

In step 3 SVD is obtained for $\bar{A}X_k = U_k^{(0)} \sum_k^{(0)} V_k^{(0)T}$ and $\bar{A}X_{k+1} = U_{k+1}^{(1)} \sum_{k+1}^{(1)} V_{k+1}^{(1)T}$ to form w_k and w_{k+1} where $w_k = X_k \upsilon_1^{(0)}$ and $w_{k+1} = X_{k+1} \upsilon_1^{(1)}$ and $\upsilon_1^{(0)}$ is the first column in $V_k^{(0)T}$ and $\upsilon_1^{(1)}$ is the first column in $V_{k+1}^{(1)T}$, so in this way we construct a monotone decreasing sequence $\{w_k\}$

such that $\|\overline{A}w_{k+1}\| \le \|\overline{A}w_k\|$ (see corollary A.4). Also SVD is used to find the singular values σ_k for $\overline{A}X_k$ and $\overline{A}X_{k+1}$ (see corollary A.3).

In fact we want to find the orthogonal complement for our current approximation w_k with respect to the desired eigenvector w of $\overline{A}$. Therefore, we are interested in seeing explicitly what happens in the subspace $w_k^{\perp}$.

The orthogonal projection of $\overline{A}$ onto the subspace is:

$$(I - w_k w_k^{\perp})\overline{A}(I - w_k w_k^{\perp})$$

Therefore, to obtain the correction vector, the following strategy is recommended:

1- Solve for t approximately from the correction equation

$$(I - w_k w_k^{\perp})\overline{A}(I - w_k w_k^{\perp})t = -(I - w_k w_k^{\perp})\varepsilon_k$$

2- Form $$x_{k+1} = \frac{t - X_k X_k^T t}{\|t - X_k X_k^T t\|}$$

If the dimension of $\mathcal{X}_{k+1}$ (where $\mathcal{X}_k$ have orthonormal basis $\{x_1, x_2 \ldots, x_k\}$ and define the matrix $X_k \in \square^{n\times k}$ as $X_k = [x_1, \ldots, x_k]$) becomes too large and approximation null space vector w is still not accurate the SNAP-JD can be restarted. Here we can use a simple restart procedure by using a single vector to restart.

From proposition A.2 we can guarantee that the smallest singular values of EA are not less than the smallest singular values of A, so any preconditioner that works for A will work for EA.

For the reason mentioned in section (3.1) we will use a right preconditioner matrix M^{-1}.

SNAP-JD(m) Algorithm:

```
Input $A, M, b, \varepsilon_{null}, k_{max}, m$

Output: $k, \|\varepsilon_k\|, \|b\|, x, w_k$

Initialization:

    Construct annihilator $E$ of $b$ as $E_{\perp}$ or $E_{|}$ and set $\bar{A} = EAM^{-1}$

Generate initial null-space vector approximation:

    Select random vector $\upsilon_0$ with $\|\upsilon_0\| = 1$, and set $\upsilon_1 = -\bar{A}\upsilon_0$. Approximately
    solve $\bar{A}t = \upsilon_1$ using GMRES(P), where p is the analytic grade. Set
    $w_1 = \frac{t + \upsilon_0}{\|t + \upsilon_0\|}$ as the initial ANS vector.

    Set null space residual vector $\varepsilon_1 = \bar{A}w_1$

    Set $x_1 = w_1$ and $X_1 = [x_1]$ then $\mathcal{X}_1 = span\{x_1\}$, and k=1

Loop for k=1,..., until $\|\varepsilon_k\| \le \varepsilon_{null}$ or k=$k_{max}$:

  Solve approximately for $t \perp w_k$ from $(I - w_k w_k^{\perp})\bar{A}(I - w_k w_k^{\perp})t = -(I - w_k w_k^{\perp})\varepsilon_k$
```

```
        using GMRES(m) (where m equals, e.g. 5 or 10)
        compute $x_{k+1} = \frac{t - X_k X_k^T t}{\|t - X_k X_k^T t\|}$
        Expand subspace $\mathcal{X}_{k+1} = \mathcal{X}_k \oplus span\{x_{k+1}\}$ and set $X_{k+1} = [X_k | x_{k+1}]$
        Extend QR-decomposition of $\bar{A}X_{k+1} = Q_{k+1}R_{k+1}$
        Obtain the SVD factorization: $R_{k+1}V_{k+1} = U_{k+1}\Sigma_{k+1}$ with smallest singular
        triplet $(\sigma_0, \upsilon_1, u_1)$
        Set $w_{k+1} = X_{k+1}\upsilon_1$ and $\varepsilon_{k+1} = \sigma_0 Q_{k+1}u_1$ with $\|\varepsilon_{k+1}\| = \sigma_0$
    End while loop
    Compute final approximate solution x of linear system:
        $x = \beta w$ , $w = M^{-1}w_k$ , $\beta = \frac{\|b\|^2}{b^T A w}$ with residual $\|r\| = |\beta| \|\varepsilon_k\|$
```

4.5 RESTARTED SNAP-JD

As mentioned in section 4.3 of method II to generate a sequence of ANS for $\bar{A}$ such that

$$\{span\{w_1, w_2, \ldots, w_k\} \mid \|\bar{A}W\| < \varepsilon, W = [w_1, w_2, \ldots, w_k]\}$$