Iterative Methods

Jayadev Naram

November 17, 2019

Contents

1 Introduction

We want to solve $Ax = b$, where $A \in \mathbb{C}^{n \times n}$ is a large non-singular matrix. To solve such a large system using Gaussian elimination is very costly and hence we turn to iterative methods for approximate solutions.

1.1 Iterative Methods

An iterative method is defined by an initial guess x_0 to the exact solution $\tilde{x} = A^{-1}b$, followed by a sequence of further approximations $\{x_i\}_{i>0}$. The error is given by $e_i = \tilde{x} - x_i$ and residual by $r_i = b - Ax_i = Ae_i$. For a given choice of x_0, an iterative method converges if $\|\tilde{x} - x_m\| \to 0$ as $m \to \infty$. An iterative method experiences finite termination if for some d we have $x_m = \tilde{x} \; \forall \; m \geq d$.

1.2 Polynomial Methods

A polynomial method is an iterative method satisfying, $e_m = P_m(A)e_0$, for some polynomial $P_m(z) = 1 - zQ_{m-1}(z)$ of degree no greater than m satisfy $P_m(0) = 1$. Then, $e_m = (I - Q_{m-1}(A)A)e_0 = e_0 - Q_{m-1}(A)r_0$. Equivalently,

$$x_m - x_0 \in \mathcal{K}_m(A, r_0),$$

where $\mathcal{K}_m(A, v) = span\{A^i v\}_{i=0}^{m-1}$ is the Krylov subspace.

1.3 Projection Methods

A projection method is an iterative method satisfying the following relation. For any m and initial guess x_0 let $\mathcal{L}_m$ be a l_m-dimensional subspace and $\mathcal{R}_m$ be a r_m-dimensional subspace. Assuming x_{m-1} exists, we define x_m to be a vector satisfying:

$$x_m - x_{m-1} \in \mathcal{R}_m, \; \tilde{x} - x_m \in \mathcal{L}_m \tag{1}$$

Note that such x_m may not exist and may not be unique. In this setting, $x_{m-1} + \mathcal{R}_m$ is called the solution space and the orthogonality condition of (1) is called the Petrov-Galerkin condition.

Now consider the specific form such x_m takes. Let $L_m(R_m)$ be $n \times l_m$ $(n \times r_m)$ matrix whose columns form a basis for $\mathcal{L}_m(\mathcal{R}_m)$. Then for some $r_m \times 1$ vector α, we have

$$x_m = x_{m-1} + R_m\alpha, \; e_m = e_{m-1} - R_m\alpha.$$

The Petrov-Galerkin condition yields, $L_m^* e_m = 0 \implies L_m^* e_{m-1} = L_m^* R_m \alpha$. If $L_m^* R_m$ is square and non-singular, then $\alpha = (L_m^* R_m)^{-1} L_m^* e_{m-1}$. Thus,

$$x_m = x_{m-1} + R_m(L_m^* R_m)^{-1} L_m^* e_{m-1},$$

$$e_m = [I - R_m(L_m^* R_m)^{-1} L_m^*] e_{m-1}.$$

Define $P_m = I - R_m(L_m^* R_m)^{-1} L_m^*$, then it easy to see that P_m is a projection matrix (How does P_m work**???**) and hence the name projection methods is appropriate. It is desirable that x_m exist and be unique for any m. Given a particular x_{m-1}, a given projection method breaks down at step m if x_m as defined by (1) does not exist or is not unique. If $l_m < r_m$ and x_m satisfying (1) exists, then such x_m cannot be unique(**???**).

Theorem 1 (**Existence/Uniquness**). *Suppose $l_m = r_m > 0$. Then breakdown occurs at step m iff $L_m^* R_m$ is singular.*

Corollary 1.1 (**Finite Termination**). *Suppose for a particular e_0 breakdown never occurs for any step of projection method, and $x_{m-1} \neq \tilde{x}$ implies $dim(\mathcal{R}_m) \gneqq dim(\mathcal{R}_{m-1})$ for any m. Then convergence is attained within n steps: $x_d = \tilde{x}$ for some $d \leq n$.*

Proof. Suppose breakdown does not occur at step m. Then by uniqueness

$$x_m = \tilde{x} \Leftrightarrow e_{m-1} \in \mathcal{R}_m,$$

$(e_m = P_m e_{m-1} = 0 \implies e_{m-1} = R_m(L_m^* R_m)^{-1} L_m^* e_{m-1})$. Since □

2 General Projection Methods

Let A $\in \mathbb{R}^{n\times n}$ and $\mathcal{K}$ *and* $\mathcal{L}$ be two m-dimensional subspaces of $\mathbb{R}^n$. A projection technique onto the subspace $\mathcal{K}$ and orthogonal to $\mathcal{L}$ with an initial guess x_0 is a process which finds an approximate solution $\tilde{x}$ by imposing the conditions that $\tilde{x}$ belong to $x_0+\mathcal{K}$ and that the new residual vector be orthogonal to $\mathcal{L}$, i.e,

$$find \quad \tilde{x} \in x_0 + \mathcal{K},\ such\ that\ \ b - A\tilde{x} \perp \mathcal{L}.$$

$$\tilde{x} = x_0 + \delta,\ \ \delta \in \mathcal{K}$$

$$(r_0 - A\delta, w) = 0,\ \forall\, w \in \mathcal{L},\ where\ r_0 = b - Ax_0.$$

Let $V = [v_1, \cdots, v_m]_{n\times m}$ *and* $W = [w_1, \cdots, w_m]_{n\times m}$ whose column-vectors form a basis of $\mathcal{K}$ and $\mathcal{L}$, respectively. Then approximate solution can be written as:

$$\tilde{x} = x_0 + Vy,$$

where y can found from the orthogonality constraint:

$$W^T AVy = W^T r_0.$$

If $W^T AV$ is non-singular, then $\tilde{x} = x_0 + V(W^T AV)^{-1}W^T r_0$.

Algorithm 1 Prototype Projection Method

```
1: repeat
2:     Select a pair of subspaces K and L
3:     Choose basis V=[v_1, ..., v_m], W=[w_1, ..., w_m] for K and L
4:     r ← b − Ax
5:     y ← (W^T AV)^{-1} W^T r
6:     x ← x + Vy
7: until Convergence
```

Non-singularity of A is not sufficient condition for non-singularity of $W^T AV$.

Proposition 1. *Let A, $\mathcal{L}$ and $\mathcal{K}$ satisfy either one of the two following conditions:*

i. A is SPD and $\mathcal{L} = \mathcal{K}$, or

ii. A is non-singular and $\mathcal{L} = A\mathcal{K}$.

Then $B = W^T AV$ is non-singular for any bases V and W of $\mathcal{K}$ and $\mathcal{L}$.

Proof. Consider case(i). Since $\mathcal{L} = \mathcal{K}$, then $W = VG$, where G is a non-singular $m \times m$ matrix. Then $B = W^T AV = G^T V^T AV$. Since A is SPD, so is $V^T AV$ and since G is non-singular, B is non-singular.
Now, consider case(ii). Since $\mathcal{L} = A\mathcal{K}$, then $W = AVG$, where G is a non-singular $m \times m$ matrix. Then $B = W^T AV = G^T(AV)^T AV$. Since A is non-singular, then $(AV)_{n\times m}$ full rank matrix and so is $(AV)^T AV$ and therefore, B is non-singular. ☐

Theorem 2. *Assume that A is SPD and $\mathcal{L} = \mathcal{K}$. Then a vector $\tilde{x}$ is the result of an (orthogonal) projection method onto $\mathcal{K}$ with the starting vector x_0 iff it minimizes the A-norm if the error over $x_0 + \mathcal{K}$, i.e, iff*

$$\tilde{x} = \underset{x \in x_0 + \mathcal{K}}{\arg\min} \|x_* - x\|_A = \underset{x \in x_0 + \mathcal{K}}{\arg\min} (A(x_* - x), x_* - x)^{\frac{1}{2}}$$

Proof. First we prove that if $\tilde{x}$ minimizes A-norm of the error, then it is the result of orthogonal projection method with x_0 onto $\mathcal{K}$. Assume columns of V to be basis vectors of $\mathcal{K}$, then the objective function can be written as:

$$\begin{aligned}
E(x) &= (A(x_* - x), x_* - x)^{\frac{1}{2}}, \qquad (x \in x_0 + \mathcal{K}) \\
\implies E(y) &= (A(x_* - x_0 - Vy), x_* - x_0 - Vy)^{\frac{1}{2}}, \;\; (y \in \mathbb{R}^m) \\
\implies E^2(y) &= (A(x_* - x_0 - Vy), x_* - x_0 - Vy), \\
&= (x_* - x_0 - Vy)^T A (x_* - x_0 - Vy), \\
&= c + 2y^T V^T (Ax_0 - Ax_*) + y^T V^T A V y, \\
&= c - 2y^T V^T (b - Ax_0) + y^T V^T A V y = f(y), \\
\frac{\partial f(y)}{\partial y} = 0 &\implies V^T(b - A(x_0 + Vy)) = 0 \\
&\implies V^T(b - A\tilde{x}) = 0 \\
&\implies b - A\tilde{x} \perp \mathcal{K}.
\end{aligned}$$

Therefore the residue of vector which minimizes A-norm of error over $x_0 + \mathcal{K}$ is orthogonal to $\mathcal{K}$, therefore it is the result of orthogonal projection method onto $\mathcal{K}$ starting with x_0. Now we prove the converse, i.e, the result of orthogonal projection method onto $\mathcal{K}$ starting with x_0 minimizes A-norm of error over $x_0 + \mathcal{K}$. We know $V^T(b - A\tilde{x}) = 0$, i.e, $(x_* - \tilde{x}, v)_A = 0 \; \forall \, v \in \mathcal{K}$.

$$\begin{aligned}
\implies \|x_* - x\|_A &= \|x_* - \tilde{x} + \tilde{x} - x\|_A, \qquad (\tilde{x}, x \in x_0 + \mathcal{K}) \\
&= \|x_* - \tilde{x}\|_A + \|\tilde{x} - x\|_A, \text{ (since } x_* - \tilde{x} \text{ is A-orthogonal to } \mathcal{K}) \\
\implies \|x_* - \tilde{x}\|_A &\leq \|x_* - x\|_A, \; \forall \, x \in x_0 + \mathcal{K}.
\end{aligned}$$

Therefore $\tilde{x}$ minimizes the A-norm of the error. □

Corollary 2.1. *Let A be an arbitrary square matrix and assume that $\mathcal{L} = A\mathcal{K}$. Then a vector $\tilde{x}$ is the result of an (oblique) projection method onto $\mathcal{K}$ orthogonally to $\mathcal{L}$ with the starting vector x_0 iff it minimizes the 2-norm of the residual vector $b - Ax$ over $x \in x_0 + \mathcal{K}$, i.e, iff*

$$\tilde{x} = \underset{x \in x_0 + \mathcal{K}}{\arg\min} \|b - Ax\|_2$$

Proposition 2. *Let $\tilde{x}$ be the approximate solution obtained from a projection process onto $\mathcal{K}$ orthogonally to $\mathcal{L} = A\mathcal{K}$, and let $\tilde{r} = b - A\tilde{x}$. Then,*

$$\tilde{r} = (I - P) r_0,$$

where P denotes the orthogonal projector onto $\mathcal{K}$.

Proof. Let $r_0 = b - Ax_0$, then

$$\begin{aligned}\tilde{r} &= b - A\tilde{x} \\ &= b - A(x_0 + \delta), \qquad (\delta \in \mathcal{K}) \\ &= r_0 - A\delta.\end{aligned}$$

By orthogonality condition we have $\tilde{r} \perp A\mathcal{K}$, i.e, $A\delta$ is the projection of r_0 onto $A\mathcal{K}$. Therefore, if P is the orthogonal projector onto $A\mathcal{K}$, then

$$Pr_0 = A\delta \implies \tilde{r} = (I - P)r_0$$

It follows from the above that $\|\tilde{r}\|_2 \leq \|r_0\|_2$. Therefore, this class of methods can be termed as **Residual Projection Methods**. $\square$

Proposition 3. *Let $\tilde{x}$ be the approximate solution obtained from an orthogonal projection process onto $\mathcal{K}$, and let $\tilde{d} = x_* - \tilde{x}$. Then,*

$$\tilde{d} = (I - P_A)d_0,$$

where P_A denotes the projector onto $\mathcal{K}$, which is orthogonal with respect to A-inner product.

Proof. Let $d_0 = x_* - x_0$ be the initial error, and let $\tilde{d} = x_* - \tilde{x}$, where $\tilde{x} = x_0 + \delta$ is the approximate solution resulting from the projection step. We know that residual of the approximate solution is orthogonal to $\mathcal{K}$, i.e, $\tilde{r} = A\tilde{d} = A(d_0 - \delta)$, $\tilde{r} \perp \mathcal{K}$.

$$\begin{aligned}&\implies (A(d_0 - \delta), w) = 0 \; \forall \; w \in \mathcal{K} \\ &\implies (d_0 - \delta, w)_A = 0 \; \forall \; w \in \mathcal{K}\end{aligned}$$

Therefore, if P_A is the projector onto $A\mathcal{K}$, which is orthogonal with respect to A-inner product, then δ is the A-orthogonal projection of d_0, i.e,

$$P_A d_0 = \delta \implies \tilde{d} = (I - P_A)d_0.$$

It follows from the above that $\|\tilde{d}\|_A \leq \|d_0\|_A$. Therefore, this class of methods can be termed as **Error Projection Methods**. $\square$

Define $\mathcal{P}_{\mathcal{K}}$ to be the orthogonal projector onto $\mathcal{K}$ and let $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ be the (oblique) projector onto $\mathcal{K}$ and orthogonally to $\mathcal{L}$. Then

$$\mathcal{P}_{\mathcal{K}}x \in \mathcal{K} \; and \; x - \mathcal{P}_{\mathcal{K}}x \perp \mathcal{K},$$

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}x \in \mathcal{K} \; and \; x - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}x \perp \mathcal{L}$$

Theorem 3. *Assume that $\mathcal{K}$ is invariant under A and the initial residue, i.e, $r_0 = b - Ax_0$ belongs to $\mathcal{K}$. Then the approximate solution obtained from any (oblique or orthogonal) projectioon method onto $\mathcal{K}$ is exact.*

Proof. An approximate solution $\tilde{x}$ is defined by

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(b - A\tilde{x}) = 0, \; where \; \tilde{x} = x_0 + \delta, \; \delta \in \mathcal{K}.$$

$$\implies \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(b - Ax_0 - A\delta) = 0$$

$$\implies \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}r_0 = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}A\delta$$

But $\mathcal{K}$ is invariant under A, then $A\delta \in \mathcal{K}$.

$$\implies r_0 = A\delta, \text{ (since } r_0 \in \mathcal{K} \text{ and } \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}A\delta = A\delta)$$

$$\implies A\tilde{x} = b$$

$\square$

Theorem 4 (**General Error Bound**). *Let* $\gamma = \|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A(I - \mathcal{P}_{\mathcal{K}})\|_2$ *and assume that* b *is a member of* $\mathcal{K}$ *and* $x_0 = 0$. *Then the exact solution* x_* *of the problem is such that*

$$\|b - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\mathcal{P}_{\mathcal{K}} x_*\|_2 \leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})x_*\|_2.$$

Proof. Since $b \in \mathcal{K}$,

$$\begin{aligned}
b - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\mathcal{P}_{\mathcal{K}} x_* &= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} b - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\mathcal{P}_{\mathcal{K}} x_* \\
&= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} (b - A\mathcal{P}_{\mathcal{K}} x_*) \\
&= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A(I - \mathcal{P}_{\mathcal{K}}) x_* \\
&= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}}) x_* \\
\implies \|b - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\mathcal{P}_{\mathcal{K}} x_*\|_2 &= \|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}}) x_*\|_2 \\
&\leq \|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A(I - \mathcal{P}_{\mathcal{K}})\|_2 \|(I - \mathcal{P}_{\mathcal{K}}) x_*\|_2 \\
\implies \|b - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\mathcal{P}_{\mathcal{K}} x_*\|_2 &\leq \gamma \|(I - \mathcal{P}_{\mathcal{K}}) x_*\|_2
\end{aligned}$$

$\square$

3 One-Dimensional Projection Methods

One-dimensional projection processes are defined when $\mathcal{K} = span\{v\}$ *and* $\mathcal{L} = span\{w\}$. In this case, the new approximation takes the form $x \leftarrow x + \alpha v$, where the orthogonality condition $r - A\delta \perp w$ yields,

$$\alpha = \frac{(r, w)}{(Av, w)}, \ where\ r = b - Ax_0.$$

3.1 Steepest Descent

The steepest descent algorithm is defined when A is SPD and $v = w = r$.

Lemma 5 (**Kantorovich inequality**). *Let* B *be any real SPD matrix and* $\lambda_1,\ \lambda_n$ *its largest and smallest eigenvalues. Then,*

$$\frac{(Bx, x)(B^{-1}x, x)}{(x, x)} \leq \frac{(\lambda_1 + \lambda_n)^2}{4\lambda_1\lambda_n},\ \forall\ x \neq 0$$

Proof. It is equivalent to prove the statement for any unit vector x. Since B is SPD, it can be diagonalized by similarity transformation with an orthogonal matrix Q, $B = Q^T DQ$.

$$(Bx, x)(B^{-1}x, x) = (Q^T DQx, x)(Q^T D^{-1}Qx, x) = (DQx, Qx)(D^{-1}Qx, Qx).$$

Define $y = Qx = (y_1, y_2, \cdots, y_n)^T,\ and\ \beta_i = {y_i}^2$. Then,

$$\lambda \equiv (Dy, y) = \sum_{i=1}^{n} \beta_i \lambda_i,\ \sum_{i=1}^{n} \beta_i = 1$$

$$\psi(y) = (D^{-1}y, y) = \sum_{i=1}^{n} \beta_i \frac{1}{\lambda_i}.$$

Note that λ is a convex combinations of eigenvalues of B. Then,

$$(Bx, x)(B^{-1}x, x) = \lambda\psi(y).$$

Noting that $f(\lambda) = 1/\lambda$ is a convex function for $x \in \mathbb{R}_{++}$, $\psi(y)$ containis all the convex combinations of $1/\lambda_i$s which is bounded above by line passing through $(\lambda_1, 1/\lambda_1)$ *and* $(\lambda_n, 1/\lambda_n)$, i.e,

$$\psi(y) \leq \frac{1}{\lambda_1} + \frac{1}{\lambda_n} - \frac{\lambda}{\lambda_1\lambda_n}.$$

$$\implies (Bx, x)(B^{-1}x, x) = \lambda\psi(y) \leq \lambda\Big(\frac{1}{\lambda_1} + \frac{1}{\lambda_n} - \frac{\lambda}{\lambda_1\lambda_n}\Big).$$

The right-hand side is maximum when $\lambda = \dfrac{\lambda_1 + \lambda_n}{2}$ yielding,

$$(Bx, x)(B^{-1}x, x) \leq \frac{(\lambda_1 + \lambda_n)^2}{4\lambda_1\lambda_n}$$

□

Algorithm 2 Steepest Descent Algorithm

Compute $r = b - Ax$ and $p = Ar$
repeat
  $\alpha \leftarrow (r, r)/(p, r)$
  $x \leftarrow x + \alpha r$
  $r \leftarrow r - \alpha p$
  Compute $p = Ar$
until Convergence

Theorem 6. *Let A be a SPD. Then, A-norms of the error vectors $d_k = x_* - x_k$ generated by the above algorithm satsify the following relation:*

$$\|d_{k+1}\|_A \leq \Big(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\Big)\|d_k\|_A,$$

and the algorithm converges for any initial guess x_0.

Proof. We know that $d_{k+1} = x_* - x_{k+1}$, but $x_{k+1} = x_k + \alpha_k r_k$.

$$\implies d_{k+1} = x_* - (x_k + \alpha_k r_k) = d_k - \alpha_k r_k.$$

Now consider,

$$\begin{aligned}
\|d_{k+1}\|_A^2 &= (d_{k+1}, d_k - \alpha_k r_k)_A \\
&= (d_{k+1}, d_k)_A - (d_{k+1}, \alpha_k r_k)_A \\
(d_{k+1}, \alpha_k r_k)_A &= (Ad_{k+1}, \alpha_k r_k) = (r_{k+1}, \alpha_k r_k), \\
&= (r_k - \alpha_k A r_k, r_k), \text{ where } \alpha_k = \frac{(r_k, r_k)}{(Ar_k, r_k)}, \\
&= (r_k, r_k) - \frac{(r_k, r_k)}{(Ar_k, r_k)}(Ar_k, r_k) = 0 = (r_{k+1}, r_k).
\end{aligned}$$

$$\implies (d_{k+1}, \alpha_k r_k)_A = 0.$$

$$
\begin{aligned}
\implies \|d_{k+1}\|_A^2 &= (d_{k+1}, d_k)_A \\
&= (d_{k+1}, Ad_k) \qquad \text{(since A is SPD)}, \\
&= (d_k - \alpha_k r_k, r_k) \\
&= (A^{-1}r_k, r_k) - \alpha_k (r_k, r_k) \\
\text{But, } \|d_k\|_A^2 = (Ad_k, d_k) &= (r_k, d_k) = (A^{-1}r_k, r_k), \\
\implies \|d_{k+1}\|_A^2 &= (A^{-1}r_k, r_k)\Big(1 - \frac{(r_k, r_k)^2}{(Ar_k, r_k)(A^{-1}r_k, r_k)}\Big),
\end{aligned}
$$

From Kantorovich inequality,

$$
\begin{aligned}
&\leq \|d_k\|_A^2 \Big(1 - \frac{4\lambda_1\lambda_n}{(\lambda_1 + \lambda_n)^2}\Big), \\
\implies \|d_{k+1}\|_A &\leq \Big(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\Big)\|d_k\|_A.
\end{aligned}
$$

□

4 Krylov Subspace Methods

We define Krylov Subspace to be

$$\mathcal{K}_m(A, v) = span\{v, Av, A^2v, \cdots, A^{m-1}v\}.$$

Then, $x = p(A)v, \ \forall\ x \in \mathcal{K}_m$, where $deg(p) < m$.

Definition 1 (Minimal Polynomial of a vector)**.** *Monic polynomial of least degree such that* $p(A)v = 0$ *is called minimal polynomial of v and degree of such polynomial is called grade*(μ).

Theorem 7. *Let* μ *be the grade of v. Then* $\mathcal{K}_\mu$ *is invariant under A and* $\mathcal{K}_\mu = \mathcal{K}_m \ \forall\ m \geq \mu$.

Proof. Since, grade of v is μ there exists a polynomial p of degree μ, such that $p(A)v = 0, \ where\ p(A) = p_0 I + p_1 A + \cdots + p_{\mu-1}A^{\mu-1} + A^\mu$.

$$\implies A^\mu v = -(p_0 I + p_1 A + \cdots + p_{\mu-1}A^{\mu-1})v \qquad (1)$$

But, $\forall\ x \in \mathcal{K}_\mu, \ x = q(A)v, \ deg(q) < \mu$, i.e,

$$
\begin{aligned}
&x = q_0 v + q_1 Av + \cdots + q_{\mu-1}A^{\mu-1}v, \ \forall\ x \in \mathcal{K}_\mu, \\
&\implies Ax = q_0 Av + q_1 A^2 v + \cdots + q_{\mu-1}A^\mu v, \\
&\text{Case 1: } q_{\mu-1} = 0, \ then\ Ax \in \mathcal{K}_\mu. \\
&\text{Case 2: } q_{\mu-1} \neq 0, \ \text{then replace } A^\mu v\ by\ (1), \ Ax \in \mathcal{K}_\mu.
\end{aligned}
$$

Therefore, $\mathcal{K}_\mu$ is invariant under A. Similarily it can be seen that $\mathcal{K}_\mu = \mathcal{K}_m$ $\forall\ m \geq \mu$. □

Corollary 7.1. $dim(\mathcal{K}_m) = min\{m, grade(v)\}$.

5 Arnoldi's Method for Linear Systems (FOM)

Arnoldi's procedure is an algorithm for building an orthogonal basis of the Krylov subspace $\mathcal{K}_m$.

Algorithm 3 Arnoldi-Modified Gram-Schmidt

```
1: Choose a vector v_1 of norm 1
2: for j = 1, 2, ..., m do
3:     Compute w_j = A v_j
4:     for i = 1, 2, ..., j do
5:         h_ij = (w_j, v_i)
6:         w_j = w_j - h_ij v_i
7:     EndDo
8:     h_{j+1,j} = ||w_j||_2.
9:     If h_{j+1,j} = 0 Stop; found an invariant subspace [v_1, ..., v_j]
10:    v_{j+1} = w_j / h_{j+1,j}
11: EndDo
```

Proposition 4. *Denote by $V_m = [v_1, v_2, \cdots, v_m]_{n\times m}$ and $\bar{H}_m$, the $(m+1)\times m$ Hessenberg matrix whose non-zero entries h_{ij} are defined by the above algorithm and by H_m the matrix obtained from $\bar{H}_m$ by removing the last row. Then,*

$$AV_m = V_m H_m + w_m e_m^T = V_{m+1}\bar{H}_m,$$
$$V_m^T A V_m = H_m.$$

Proof. From lines 6,8 we have, $w_j = Av_j - h_{ij}v_i$ and $w_j = v_{j+1}h_{j+1,j}$.

$$\implies Av_j = \sum_{i=1}^{j+1} h_{ij}v_i \implies AV_m = V_m H_m + w_m e_m^T = V_{m+1}\bar{H}_m.$$

Since V_m^T is orthogonal, we get $V_m^T A V_m = H_m$.

□

Given an initial guess x_0 to the original linear system $Ax = b$, we now consider an orthogonal projection method which takes $\mathcal{L} = \mathcal{K} = \mathcal{K}_m(A, r_0)$, with

$$\mathcal{K}_m(A, r_0) = span\{r_0, Ar_0, A^2 r_0, \cdots, A^{m-1}r_0\},$$

in which $r_0 = b - Ax_0$. This method seeks an approximate solution x_m from the affine subspace $x_0 + \mathcal{K}_m$ of dimension m by imposing the following orthogonality constraint:

$$b - Ax_m \perp \mathcal{K}_m.$$

If $v_1 = r_0/\|r_0\|_2$ in Arnoldi's method, and we set $\beta = \|r_0\|_2$, then

$$V_m^T A V_m = H_m, \; V_m^T r_0 = V_m^T(\beta v_1) = \beta e_1.$$

As a result, the approximate solution using the above m-dimensional subspaces is given by:

$$x_m = x_0 + V_m y_m,$$

where y_m can be found by imposing orthogonality constraint that

$$V_m^T(b - Ax_m) = 0 \implies y_m = H_m^{-1}(\beta e_1).$$

Algorithm 4 Full Orthogonalization Method (FOM)

```
Compute r_0 = b - Ax_0, β = ||r_0||_2, and v_1 = r_0/β
Define the m × m matrix H_m = {h_ij}_{i,j=1,2,··· ,m}; Set H_m = 0
for j = 1, 2, ··· , m do
    Compute w_j = Av_j
    for i = 1, 2, ··· , j do
        h_ij = (w_j, v_i)
        w_j = w_j - h_ij v_i
    EndDo
    h_{j+1,j} = ||w_j||_2. If h_{j+1,j} = 0 Stop
    v_{j+1} = w_j / h_{j+1,j}
EndDo
Compute y_m = H_m^{-1} β e_1 and x_m = x_0 + V_m y_m
```

Proposition 5. *The residual vector of the approximate solution x_m computed by the FOM Algorithm is such that*

$$r_m = b - Ax_m = -h_{m+1,m} e_m^T y_m v_{m+1}$$

and, therefore,

$$\|r_m\|_2 = \|b - Ax_m\|_2 = h_{m+1,m} |e_m^T y_m|.$$

Proof.

$$\begin{aligned} r_m &= b - Ax_m \\ &= b - Ax_0 - AV_m y_m \\ &= r_0 - (V_m H_m + w_m e_m^T) y_m \\ &= r_0 - V_m H_m (H_m^{-1} \beta e_1) - w_m e_m^T y_m \\ &= r_0 - V_m V_m^T r_0 - h_{m+1,m} e_m^T y_m v_{m+1} = -h_{m+1,m} e_m^T y_m v_{m+1}. \end{aligned}$$

□

5.1 Variation 1: Restarted FOM

Algorithm 5 Restarted FOM (FOM(m))

```
1: Compute r_0 = b - Ax_0, β = ||r_0||_2, and v_1 = r_0/β
2: Generate V_m and H_m using Arnoldi algorithm starting with v_1.
3: Compute y_m = H_m^{-1} β e_1 and x_m = x_0 + V_m y_m. If satisfied then Stop.
4: Set x_0 = x_m and go to 1.
```

5.2 Variation 1: IOM and DIOM

A formula can be developed whereby the current approximate solution x_m can be computed from the previous approximation x_{m-1} and a small number vectors are updated at each step. This progressive formulation of the solution leads to an algorithm termed as Direct IOM (DIOM).

Algorithm 6 Incomplete Orthogonalization Method (IOM)

```
Compute r_0 = b − A x_0,  β = ||r_0||_2,  and v_1 = r_0/β
Define the m × m matrix H_m = {h_ij}_{i,j=1,2,···,m}; Set H_m = 0
for j = 1, 2, ···, m do
    Compute w_j = A v_j
    for i = max{1, j − (k − 1)}, 2, ···, j do
        h_ij = (w_j, v_i)
        w_j = w_j − h_ij v_i
    EndDo
    h_{j+1,j} = ||w_j||_2. If h_{j+1,j} = 0 Stop
    v_{j+1} = w_j / h_{j+1,j}
EndDo
Compute y_m = H_m^{-1} β e_1 and x_m = x_0 + V_m y_m
```

The Hessenberg matrix obtained from IOM has a band structure with bandwidth $k+1$, i.e,

$$\begin{aligned} H_m &= \begin{pmatrix} h_{11} & h_{12} & h_{13} & & \\ h_{21} & h_{22} & h_{23} & h_{24} & \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \end{pmatrix} = L_m U_m \\ &= \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ & l_{32} & 1 & & \\ & & l_{43} & 1 & \\ & & & l_{54} & 1 \end{pmatrix} \times \begin{pmatrix} u_{11} & u_{12} & u_{13} & & \\ & u_{22} & u_{23} & u_{24} & \\ & & u_{33} & u_{34} & u_{35} \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{pmatrix} \end{aligned}$$

The approximate solution then is given by

$$x_m = x_0 + V_m U_m^{-1} L_m^{-1} (\beta e_1).$$

Define $P_m \equiv V_m U_m^{-1}$ *and* $z_m = L_m^{-1}(\beta e_1)$, we have $x_m = x_0 + P_m z_m$. Because of the structure of U_m, P_m can be updated easily. Indeed, equating the last columns of the matrix relation $P_m U_m = V_m$ yields,

$$\sum_{i=m-k+1}^{m} u_{im} p_i = v_m \implies p_m = \frac{1}{u_{mm}} \left(v_m - \sum_{i=m-k+1}^{m-1} u_{im} p_i \right).$$

Therefore, p_m can be computed using previous $p_i's$ and v_m. In addition, due to the structure of L_m, we have compute z_m by,

$$z_m = \begin{bmatrix} z_{m-1} \\ \zeta_m \end{bmatrix}, \text{ where } \zeta_m = -l_{m,m-1}\zeta_{m-1}.$$

Now, the approximate solution is,

$$x_m = x_0 + \begin{bmatrix} P_{m-1} & p_m \end{bmatrix} \begin{bmatrix} z_{m-1} \\ \zeta_m \end{bmatrix} = x_0 + P_{m-1} z_{m-1} + p_m \zeta_m.$$

Noting that $x_{m-1} = P_{m-1}z_{m-1}$, x_m can be updated as follows:

$$x_m = x_{m-1} + \zeta_m p_m.$$

This gives the following algorithm, called **Incomplete Orthogonalization Method**(DIOM).

Algorithm 7 Direct Incomplete Orthogonalization Method (DIOM)

Choose x_0 and compute $r_0 = b - Ax_0,\ \beta = \|r_0\|_2,\ and\ v_1 = r_0/\beta$
for $m = 1, 2, \cdots$, until convergence **do**
  Compute $w_m = Av_m$
  for $i = max\{1, m-k+1\}, 2, \cdots, m$ **do**
    $h_{im} = (w_m, v_i)$
    $w_m = w_m - h_{im}v_i$
  $h_{m+1,m} = \|w_m\|_2$. If $h_{m+1,m} = 0$ Stop
  $v_{m+1} = w_m/h_{m+1,m}$
  Update the LU factorization of H_m, i.e, obtain the last column
    U_m using the previous k pivots. If $u_{mm} = 0$ Stop.
  $\zeta_m = \beta$ if $m = 1$ else $-l_{m,m-1}\zeta_{m-1}$
  $p_m = u_{mm}^{-1}\left(v_m - \sum_{i=m-k+1}^{m-1} u_{im}p_i\right)$(for $i \leq 0$ set $u_{im}p_i \equiv 0$)
  $x_m = x_{m-1} + \zeta_m p_m$
EndDo

Remark. *Observe that $V_m^T AV_m = H_m$ is still valid because the orthogonality properties were not used to derive this relation. As a consequence the following result is also valid,*

$$\begin{aligned} r_m = b - Ax_m &= -h_{m+1,m}e_m^T y_m v_{m+1} \\ \implies \|b - Ax_m\|_2 &= h_{m+1,m}|e_m^T y_m| \\ \textit{But, } y_m = H_m^{-1}(\beta e_1) &= U_m^{-1}z_m \implies e_m^T y_m = \zeta_m/u_{mm} \\ \implies \|b - Ax_m\|_2 &= h_{m+1,m}\left|\frac{\zeta_m}{u_{mm}}\right| \end{aligned}$$

Since the residual vectors is a scalar multiple of v_{m+1} and since the v_i's are no longer orthogonal, IOM and DIOM are not orthogonal projection techniques. They can however be viewed as oblique projection techniques onto $\mathcal{K}_m$ orthogonally to an artificially constructed subspace.

Proposition 6. *IOM and DIOM are mathematically equivalent to projection process onto $\mathcal{K}_m$ and orthogonally to*

$$\mathcal{L}_m = span\{z_1, z_2, \cdots, z_m\},$$

$$where\ z_i = v_i - (v_i, v_{m+1})v_{m+1},\ i = 1, 2, \cdots, m.$$

Proof. From the construction of $\mathcal{L}_m$, v_{m+1} is orthogonal to $\mathcal{L}_m$ and we know the final residue r_m is a scalar multiple of v_{m+1}, hence the approximate solution $x_m \in \mathcal{K}_m$ and residue vector $r_m \perp \mathcal{L}_m$. □

6 Symmetric Lanczos Algorithm

The symmetric lanczos algorithm can be viewed as a simplification of Arnoldi's method for the particular case of symmetric matrix. When A is symmetric, then the Hessenberg matrix H_m will become symmetric tridiagonal. The standard notation used to descsribe the Lanczos algorithm is obtained by setting

$$\alpha_j = h_{jj},\ \beta_j = h_{j-1,j},$$

and if T_m denotes the resulting H_m matrix, it is of the form,

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & . & . & . & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}.$$

This leads to the following form of Modified Gram-Schmidt variant of Arnoldi's method:

Algorithm 8 Lanczos Method for Linear Systems

```
Compute r_0 = b − Ax_0, β = ||r_0||_2, and v_1 = r_0/β
Set β_1 = 0 and v_0 = 0
for j = 1, 2, ··· , m do                         ▷ Orthogonalization Procedure
    w_j = Av_j − β_j v_{j−1}
    α_j = (w_j, v_j)
    w_j = w_j − α_j v_j
    β_{j+1} = ||w_j||_2. If β_{j+1} = 0 then Stop
    v_{j+1} = w_j/β_{j+1}
EndDo
Set T_m = tridiag(β_i, α_i, β_{i+1}), and V_m = [v_1, ··· , v_m].
Compute y_m = H_m^{−1} β e_1 and x_m = x_0 + V_m y_m
```

7 Conjugate Gradient

The conjugate gradient algorithm can be derived from the Lanczos algorithm in the same way DIOM was derived from IOM. Infact, the conjugate gradient algorithmm can be viewed as a variation of DIOM for the case when A is symmetric.

First write the LU factorization of T_m as $T_m = L_m U_m$. The matrix L_m is unit lower bidiagonal and U_m is unit upper bidiagonal matrix. Thus the factorization of T_m is of the form

$$T_m = \begin{pmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & . & . & & \\ & & \lambda_{m-1} & 1 & \\ & & & \lambda_m & 1 \end{pmatrix} \times \begin{pmatrix} \eta_1 & \beta_2 & & & \\ & \eta_2 & \beta_3 & & \\ & & . & . & \\ & & & \eta_{m-1} & \beta_m \\ & & & & \eta_m \end{pmatrix}.$$

The approximate solution is then given by,

$$x_m = x_0 + V_m U_m^{-1} L_m^{-1} (\beta e_1) = x_0 + P_m z_m.$$

As for DIOM, p_m, the last column of P_m, can be computed from the previous $p_i's\ and\ v_m$ by the simple update

$$p_m = \eta_m^{-1} [v_m - \beta_m p_{m-1}].$$

Note that β_m is a scalar computed from the Lanczos algorithm, while η_m results from the m-th Gaussian elimination step on the tridiagonal matrix, i.e, $\lambda_m = \frac{\beta_m}{\eta_{m-1}},\ \eta_m = \alpha_m - \lambda_m \beta_m$. In addition, following again what has been shown for DIOM, $z_m = \begin{bmatrix} z_{m-1} \\ \zeta_m \end{bmatrix},\ where\ \zeta_m = -\lambda_m \zeta_{m-1}$. As a result, x_m can be updated at each step as follows:

$$x_m = x_{m-1} + \zeta_m p_m.$$

This gives the following algorithm, which we call as direct version of Lanczos algorithm for linear systems.

Algorithm 9 D-Lanczos

Choose x_0 and compute $r_0 = b - Ax_0,\ \zeta_1 = \beta = \|r_0\|_2,\ and\ v_1 = r_0/\beta$
$\lambda_1 = \beta_1 = 0,\ p_0 = 0$
for $m = 1, 2, \cdots,$ until convergence **do**
  Compute $w_m = Av_m - \beta_m v_{m-1}\ and\ \alpha_m = (w, v_m)$
  If $m > 1$ then compute $\lambda_m = \frac{\beta_m}{\eta_{m-1}}$ and $\zeta_m = -\lambda_m \zeta_{m-1}$
  $\eta_m = \alpha_m - \lambda_m \beta_m$
  $p_m = \eta_m^{-1} [v_m - \beta_m p_{m-1}]$
  $x_m = x_{m-1} + \zeta_m p_m$
  If x_m has converged then Stop
  $w = w - \alpha_m v_m$
  $\beta_{m+1} = \|w\|_2,\ v_{m+1} = w/\beta_{m+1}$
EndDo

Observe that the residual vector for this algorithm is in the direction of v_{m+1}. Therefore, the residual vectors are orthogonal to each other as in FOM. Likewise, the vectors p_i are A-orhogonal or conjugate to each other.

Proposition 7. *Let $r_m = b - Ax_m,\ and\ p_m, m = 0, 1, \cdots,$ be the residual vectors and auxiliary vectors produced by D-Lanczos algorithm. Then,*

i) Each residual vector r_m is such that $r_m = \sigma_m v_{m+1}$, where σ_m is a certain scalar. As a result, the residual vectors are orthogonal to each other.

ii) The auxiliary vectors p_i form an A-conjuagte set, i.e,

$$(Ap_i, p_j) = 0,\ for\ i \neq j.$$

Proof. The first part is immediate consequence of the following relation:

$$r_m = b - Ax_m = -h_{m+1,m} e_m^T y_m v_{m+1}$$

Now we prove the second part. Consider the matrix $P_m^T A P_m$,

$$P_m^T A P_m = U_m^{-T} V_m^T A V_m U_m^{-1} = U_m^{-T} T_m U_m^{-1} = U_m^{-T} L_m$$

Now we observe $U_m^{-T} L_m$ is a lower triangular matrix which is also symmetric since it is equal to the symmetric matrix $P_m^T A P_m$. Therefore it must be diagonal. □

A consequence of the above proposition is that a version of the algorithm can be derived by imposing the orthogonality and conjugacy conditions. This gives the Conguate Gradient algorithm which we now derive.

Remark. *Inorder to conform with the standard notation used in the literature to describe the algorithm, the indexing of the p vectors now begins at zero instead of one as was done so far.*

The vector $x_{j+1}, r_{j+1}, p_{j+1}$ can be expressed as follows:

$$x_{j+1} = x_j + \alpha_j p_j, \quad r_{j+1} = r_j - \alpha_j A p_j, \quad p_{j+1} = r_{j+1} + \beta_j p_j.$$

It can be seen that $(Ap_j, r_j) = (Ap_j, p_j + \beta_j p_{j-1}) = (Ap_j, p_j)$. Now imposing constraints,

1. Orthogonality Conditions on r_i's gives $(r_j - \alpha_j A p_j, r_j) = 0$, as a result,

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, r_j)} = \frac{(r_j, r_j)}{(Ap_j, p_j)}$$

2. Conjugacy Conditions on p_i's gives $(p_{j+1}, Ap_j) = 0$. But,

$$(p_{j+1}, Ap_j) = (r_{j+1} + \beta_j p_j, Ap_j) = -\frac{1}{\alpha_j}(r_{j+1} + \beta_j p_j, r_{j+1} - r_j)$$

$$(p_{j+1}, Ap_j) = 0 \implies \beta_j = \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$$

It is important to note that the scalar α_j, β_j in this algorithm and D-Lanczos are different.

Putting these relation together gives the following algorithm.

Algorithm 10 Conjugate Gradient

Choose x_0 and compute $r_0 = b - Ax_0$, $p_0 = r_0$.
for $j = 0, 1, 2, \cdots$, until convergence **do**
  $\alpha_j = (r_j, r_j)/(Ap_j, p_j)$
  $x_{j+1} = x_j + \alpha_j p_j$
  $r_{j+1} = r_j - \alpha_j A p_j$
  $\beta_j = (r_{j+1}, r_{j+1})/(r_j, r_j)$
  $p_{j+1} = r_{j+1} + \beta_j p_j$
EndDo

Remark (LDL^T Factorization of T_m)**.** *We have already seen the LU decomposition of T_m:*

$$T_m = \begin{pmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & \cdot & \cdot & & \\ & & \lambda_{m-1} & 1 & \\ & & & \lambda_m & 1 \end{pmatrix} \times \begin{pmatrix} \eta_1 & \beta_2 & & & \\ & \eta_2 & \beta_3 & & \\ & & \cdot & \cdot & \\ & & & \eta_{m-1} & \beta_m \\ & & & & \eta_m \end{pmatrix},$$

where

$$\eta_1 = \alpha_1,\ \lambda_k = \frac{\beta_k}{\eta_{k-1}},\ \eta_k = \alpha_k - \lambda_k\beta_k,\ k = 2, 3, \cdots, m.$$

Notice here that U can be further factorized as $U = DL^T$ such that $D = diag(\eta_1, \eta_2, \cdots, \eta_m)$. Therefore we arrive at the LDL^T decomposition of T_m to be $T_m = L_m D_m L_m^T$, i.e,

$$T_m = \begin{pmatrix} 1 & & & \\ \lambda_2 & 1 & & \\ & \cdot & \cdot & \\ & & \lambda_m & 1 \end{pmatrix} \begin{pmatrix} \eta_1 & & & \\ & \eta_2 & & \\ & & \cdot & \\ & & & \eta_m \end{pmatrix} \begin{pmatrix} 1 & & & \\ \lambda_2 & 1 & & \\ & \cdot & \cdot & \\ & & \lambda_m & 1 \end{pmatrix}^T.$$

Proposition 8. *Let $A_m = (P_m^T P_m)^{-1}$, then $A_{m+1} = \begin{bmatrix} \tilde{A}_m & \lambda_{m+1}e_m \\ \lambda_{m+1}e_m^T & 1 \end{bmatrix}$, where $\tilde{A}_m = A_m + \lambda_{m+1}^2 e_m e_m^T$.*

Proof. First notice that from the above factorization of L_m that,

$$P_m = V_m L_m^{-T} \implies A_m = L_m^T L_m.$$

Then

$$\begin{aligned} A_m &= \begin{pmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & \cdot & \cdot & & \\ & & \cdot & 1 & \\ & & & \lambda_m & 1 \end{pmatrix} \begin{pmatrix} 1 & \lambda_2 & & & \\ & 1 & \cdot & & \\ & & \cdot & \cdot & \\ & & & 1 & \lambda_m \\ & & & & 1 \end{pmatrix}^T \\ &= \begin{pmatrix} 1+\lambda_2^2 & \lambda_2 & & & \\ \lambda_2 & 1+\lambda_3^2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & 1+\lambda_m^2 & \lambda_m \\ & & & \lambda_m & 1 \end{pmatrix} \end{aligned}$$

Therefore, $A_m = (P_m^T P_m)^{-1}$ is tridiagonal and A_{m+1} can constructed from A_m as follows: $A_{m+1} = \begin{bmatrix} \tilde{A}_m & \lambda_{m+1}e_m \\ \lambda_{m+1}e_m^T & 1 \end{bmatrix}$, where $\tilde{A}_m = A_m + \lambda_{m+1}^2 e_m e_m^T$. $\square$

Proposition 9. *The r_j vectors obtained in the Conjugate Gradient algorithm follow the following recurrence relation:*

$$Ar_j = \hat{\beta}_j r_{j-1} + (\alpha_{j+1}) r_j + \hat{\beta}_{j+1} r_{j+1}.$$

Proof. We know, $r_m = b - Ax_m = c_m v_{m+1} = \{-\beta_{m+1}(e_m^T y_m)\}v_{m+1}$, where $y_m = T_m^{-1}(\beta_0 e_1)$. Then $e_m^T y_m = e_m^T L_m^{-T} D_m^{-1} L_m^{-1}(\beta_0 e_1)$. Now we evaluate the value of $e_m^T y_m$. It can be seen that

$$L_m^{-1} = \begin{pmatrix} 1 & & & & & \\ -\lambda_2 & 1 & & & & \\ \lambda_2\lambda_3 & -\lambda_3 & 1 & & & \\ \vdots & & & \ddots & & \\ & & & & 1 & \\ \hat{l} & -\hat{l}/\lambda_2 & \dots & & -\lambda_m & 1 \end{pmatrix}, \text{ where } \hat{l} = (-1)^{m-1}\prod_{i=2}^{m}\lambda_i.$$

Each entry of L_m^{-1} can be written as,

$$(L_m^{-1})_{i,j} = \begin{cases} (-1)^{i-j}\prod\limits_{k=j+1}^{i}\lambda_k, & \text{for } i > j \\ 1, & \text{for } i = j \\ 0, & \text{for } i < j \end{cases}$$

Now notice that, $L_m^{-1}e_1$ will give the first column of L_m^{-1} and similarly $e_m^T L_m^{-T} = (L_m^{-1}e_m)^T = e_m^T$ and hence $e_m^T D_m^{-1} = \frac{1}{\eta_m}e_m^T$. Then

$$c_m = -\frac{\beta_0\beta_{m+1}}{\eta_m}e_m^T L_m^{-T}e_1 = -\frac{\beta_0\beta_{m+1}}{\eta_m}(L_m^{-1})_{m,1} = \frac{(-1)^m}{\eta_m}\beta_0\beta_{m+1}\prod_{i=2}^{m}\lambda_i.$$

Therefore, $r_m = c_m v_{m+1}$, where $c_m = \dfrac{(-1)^m}{\eta_m}\beta_0\beta_{m+1}\prod\limits_{i=2}^{m}\lambda_i$.

From Lanczos algorithm we know, $AV_m = V_m T_m + \beta_{m+1}v_{m+1}e_m^T$. On compare the last column of the matrices on both sides of the equation we get,

$$Av_m = \beta_m v_{m-1} + \alpha_m v_m + \beta_{m+1}v_{m+1}.$$

Raising the index of the terms in the above equation we get,

$$Av_{m+1} = \beta_{m+1}v_m + \alpha_{m+1}v_{m+1} + \beta_{m+2}v_{m+2}.$$

On substituting r_m in it leads to,

$$\begin{aligned} A\frac{r_m}{c_m} &= \beta_{m+1}\frac{r_{m-1}}{c_{m-1}} + \alpha_{m+1}\frac{r_m}{c_m} + \beta_{m+2}\frac{r_{m+1}}{c_{m+1}} \\ \implies Ar_m &= \left(\frac{\beta_{m+1}c_m}{c_{m-1}}\right)r_{m-1} + \alpha_{m+1}r_m + \left(\frac{\beta_{m+2}c_m}{c_{m+1}}\right)r_{m+1}. \end{aligned}$$

□

8 Convergence Analysis

Here we show the convergence of CG Algorithm using Chebyshev polynomials.

8.1 Real Chebyshev Polynomials

Definition 2. *The Chebyshev polynomial of the first kind of degree k is defined by:*

$$C_k(t) = \begin{cases} cos[k\,cos^{-1}(t)], & for\ |t| \leq 1 \\ cosh[k\,cosh^{-1}(t)], & otherwise. \end{cases}$$

It can be seen that $C_0(t) = 1,\ C_1(t) = t$, can be easily extended by the following trigonometric relation

$$cos[(k+1)\theta] + cos[(k-1)\theta] = 2cos\theta\, cosk\theta.$$

This also shows the important three-term recurrence relation

$$C_{k+1}(t) = 2tC_k(t) - C_{k-1}(t).$$

Proposition 10. *If* $|t| > 1$, *then* $C_k(t) = \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k + \left(t+\sqrt{t^2-1}\right)^{-k}\right]$

Proof. We know, $C_k(t) = cosh[k\,cosh^{-1}(t)], |t| > 1,\ cosh\,\theta = \frac{1}{2}\left(e^{\theta} + e^{-\theta}\right)$. Then, $C_k(t) = \frac{1}{2}\left[(e^{\theta})^k + (e^{\theta})^{-k}\right]$, where $\theta = cosh^{-1}(t)$, i.e, $t = cosh\,\theta$

$$\begin{aligned} &\implies t = \frac{1}{2}\left(e^{\theta} + e^{-\theta}\right) \implies e^{2\theta} - 2te^{\theta} + 1 = 0 \\ &\implies e^{\theta} = t \pm \sqrt{t^2-1} \end{aligned}$$

Notice that $t + \sqrt{t^2-1} = \frac{1}{t-\sqrt{t^2-1}}$. Then, for $|t| > 1$,

$$C_k(t) = \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k + \left(t+\sqrt{t^2-1}\right)^{-k}\right] \gtrapprox \frac{1}{2}\left(t+\sqrt{t^2-1}\right)^k.$$

□

In what follows we denote by $\mathbb{P}_k$ as the set of all polynomials of degree k.

Remark. *The Chebyshev polynomials are polynomials with the largest possible leading coefficient whose absolute value on the interval* $[-1, 1]$ *is bounded by 1.*

Theorem 8. *Let* $[\alpha, \beta]$ *be a non-empty interval in* $\mathbb{R}$ *and let* γ *be any real scalar outside* $[\alpha, \beta]$. *Then,*

$$\hat{C}_k(t) = \frac{C_k\left(1 + 2\frac{t-\beta}{\beta-\alpha}\right)}{C_k\left(1 + 2\frac{\gamma-\beta}{\beta-\alpha}\right)} = \underset{p\in\mathbb{P}_k,\, p(\gamma)=1}{\arg\min}\ \max_{t\in[\alpha,\beta]} |p(t)|.$$

Remark. *If* $\gamma \leq \alpha$, *the absolute values are needed in the denominators and exchanging the roles of* α *and* β, *i.e,*

$$\hat{C}_k(t) = \frac{C_k\left(1 + 2\frac{\alpha-t}{\beta-\alpha}\right)}{C_k\left(1 + 2\frac{\alpha-\gamma}{\beta-\alpha}\right)}.$$

We can further state following which is the result of the above remark and theorem.

Corollary 8.1. $\underset{p\in\mathbb{P}_k,\, p(\gamma)=1}{\arg\min}\ \max_{t\in[\alpha,\beta]} |p(t)| = \dfrac{1}{|C_k(1 + 2\frac{\gamma-\beta}{\beta-\alpha})|}$

8.2 Convergence of CG Algorithm

Lemma 9. *Let x_m be the approximate solution obtained from the m-th step of the CG algorithm, and let $d_m = x_* - x_m$ where x_* is the exact solution. Then, $x_m = x_0 + q_m(A)r_0$, where q_m is a polynomial of degree $m-1$ such that*

$$\|(I - Aq_m(A))d_0\|_A = \min_{q\in\mathbb{P}_{m-1}} \|(I - Aq(A))d_0\|_A.$$

Proof. Consider the following objective function in the polynomial q

$$\|(I - Aq(A))d_0\|_A = \|d_0 - Aq(A)d_0\|_A = \|x_* - (x_0 + Aq(A)r_0)\|_A = \|x_* - x\|_A.$$

And also observe that $\forall\, x \in x_0 + \mathcal{K}_m(A, r_0)$, $x = x_0 + q(A)r_0, q \in \mathbb{P}_{m-1}$. Therefore minimizing x over $x_0 + \mathcal{K}_m(A, r_0)$ and q over $\mathbb{P}_{m-1}$ are equivalent. We know that $x_m = \underset{x\in x_0+\mathcal{K}}{\arg\min}\|x_* - x\|_A$. Hence, $q_m = \underset{q\in\mathbb{P}_{m-1}}{\arg\min}\|(I - Aq(A))d_0\|_A$. $\square$

Theorem 10. *Let $\eta = \dfrac{\lambda_n}{\lambda_1 - \lambda_n}$ and $\kappa = \dfrac{\lambda_1}{\lambda_n}$, then*

$$\|x_* - x_m\|_A \le \frac{\|x_* - x_0\|_A}{C_m(1+2\eta)} \le 2\left[\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right]^m \|x_* - x_0\|_A.$$

Proof. From the previous lemma, it is known that $\|x_* - x_m\|_A$ is the minimum of the error over polynomials $r(t) = 1 - tq(t)$ which take the value one at 0, i.e,

$$\|x_* - x_m\|_A = \min_{r\in\mathbb{P}_m,\, r(0)=1} \|r(A)d_0\|_A.$$

Let $\lambda_i, i = 1, 2, \cdots, n$ are the eigenvalues of A, and $\xi_i, i = 1, 2, \cdots, n$ are the components of the initial error d_0 in the eigenbasis. Let eigen decomposition of $A = U\Lambda U^T$, where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$, then $d_0 = U\xi$, $\xi = [\xi_1\ \xi_2\ \cdots\ \xi_n]^T$ and $r(A) = Ur(\Lambda)U^T$. Then,

$$\begin{aligned}
\|r(A)d_0\|_A^2 &= (Ar(A)d_0, r(A)d_0) = d_0^T r(A)A\, r(A)d_0\\
&= \xi^T U^T U r(\Lambda)U^T U\Lambda U^T U r(\Lambda)U^T U\xi\\
&= \xi^T \Lambda r(\Lambda)^2\xi = \sum_{i=1}^n \lambda_i r(\lambda_i)^2\xi_i^2 \le \max_i (r(\lambda_i)^2)\|d_0\|_A^2\\
\implies \|r(A)d_0\|_A^2 &\le \max_{\lambda\in[\lambda_n,\lambda_1]} (r(\lambda_i)^2)\|d_0\|_A^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|x_* - x_m\|_A &\le \min_{r\in\mathbb{P}_m,\, r(0)=1}\ \max_{\lambda\in[\lambda_n,\lambda_1]} |r(\lambda_i)|\|d_0\|_A\\
&= \frac{\|x_* - x_0\|_A}{C_m(1+2\eta)}.
\end{aligned}$$

We know, $C_m(t) \ge \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^m$, then

$$C_m(1+2\eta) \ge \frac{1}{2}\left(1 + 2\eta + 2\sqrt{\eta(\eta-1)}\right)^m.$$

Now notice that

$$1 + 2\eta + 2\sqrt{\eta(\eta-1)} = (\sqrt{\eta} + \sqrt{\eta+1})^2 = \frac{(\sqrt{\lambda_n} + \sqrt{\lambda_1})^2}{\lambda_1 - \lambda_n} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$$

$$\text{Therefore, } \|x_* - x_m\|_A \leq 2\left[\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right]^m \|x_* - x_0\|_A.$$

□

9 Deflated-CG Algorithm

9.1 Deflated Lanczos algorithm

A $\in \mathbb{R}^{n\times n}$ is SPD and define $W = [w_1 \ w_2 \ \cdots \ w_k]$, $w_i \in \mathbb{R}^n$ are linearly independent vectors. Let $v_1 \in \mathbb{R}^n$ be a unit vector such that $W^T v_1 = 0$.

Proposition 11. *Let W and A be defined as above, then W^TAW is SPD.*

Proof. Let $x \in \mathbb{R}^k$. We know that the columns of W are linearly independent, hence $Wx = 0 \Leftrightarrow x = 0$. Let $y = Wx$, then $x^TW^TAWx = y^TAy \geq 0$ (since A is SPD). □

We want to generate a sequence $\{v_j\}_{j=1,2,\cdots}$ of vectors such that

$$v_{j+1} \perp \text{span}\{W, v_1, v_2, \cdots, v_j\}, \ \|v_{j+1}\|_2 = 1.$$

To obtain such a sequence, we apply Lanczos algorithm to the following auxiliary matrix:

$$B = A - AW(W^TAW)^{-1}W^TA,$$

with the given initial vector v_1. Then we get a sequence $\{v_j\}_{j=1,2,\cdots}$ which satisfies

$$BV_j = V_jT_j + \sigma_{j+1}v_{j+1}e_j^T, \text{ where } T_j = \begin{pmatrix} \rho_1 & \sigma_2 & & & \\ \sigma_2 & \rho_2 & \sigma_3 & & \\ & . & . & . & \\ & & \sigma_{j-1} & \rho_{j-1} & \sigma_j \\ & & & \sigma_j & \rho_j \end{pmatrix}$$

It is guaranteed by the Lanczos algorithm that the vectors v_j are orthogonal to each other. By comparing the last column on the both sides of the above equation, we get

$$\sigma_{j+1}v_{j+1} = Bv_j - \sigma_j v_{j-1} - \rho_j v_j.$$

It can also be seen that $W^TB = 0$.

Proposition 12. *Let W and v_{j+1} be defined as above. Then* $W^Tv_{j+1} = 0 \ \forall \ j = 1, 2, \cdots$.

Proof. We prove by induction on j. It is clear from the definition that $W^Tv_1 = 0$. Now assume that the propsition is true till $j-1$, i.e, $W^Tv_{k+1} = 0 \ \forall \ k = 1, 2, \cdots, j$. Now we prove for $k = j+1$.

We know, $\sigma_{j+1}v_{j+1} = Bv_j - \sigma_j v_{j-1} - \rho_j v_j$, then

$$\Longrightarrow \sigma_{j+1}W^Tv_{j+1} = W^TBv_j - \sigma_j W^Tv_{j-1} - \rho_j W^Tv_j = 0.$$

□

Therefore, we arrive at the following algorithm:

Algorithm 11 Deflated Lanczos Algorithm

Choose k linearly independent vectors $w_1, w_2, \cdots, w_k$ and
    define $W = [w_1 \; w_2 \; \cdots \; w_k]$
Choose an initial vector v_1 such that $W^T v_1 = 0$ and $\|v_1\|_2 = 1$.
for $j = 1, 2, \cdots, m$ **do**
    Solve $W^T AW \hat{v}_j = W^T A v_j$ for $\hat{v}_j$
    $z_j = Av_j - AW\hat{v}_j$
    $\rho_j = v_j^T z_j$
    $\widehat{v_{j+1}} = z_j - \sigma_j v_{j-1} - \rho_j v_j$
    $\sigma_{j+1} = \|\widehat{v_{j+1}}\|_2$; If $\sigma_{j+1} = 0$ Exit.
    $v_{j+1} = \widehat{v_{j+1}}/\sigma_{j+1}$
EndDo

9.2 Deflated-CG algorithm

We want to solve $Ax = b$, A is SPD. Based on deflated Lanczos algorithm we wish to derive a projection method similar to CG-algorithm.

Assume x_0 is the initial guess and initial residue $r_0 = b - Ax_0$ such that $r_0 \perp W$. Now set $v_1 = r_0/\|r_0\|_2$ and define $\mathcal{K}_{k,j}(A, W, r_0) \equiv \text{span}\{W, V_j\}$. At j-th step of the projection, we seek for the approximate solution $x_j \in x_0 + \mathcal{K}_{k,j}(A, W, r_0)$ and $r_j = b - Ax_j \perp \mathcal{K}_{k,j}(A, W, r_0)$.

Lemma 11. *If x_j and r_j satisfy the above conditions, then $r_j = c_j v_{j+1}$, for some scalar c_j. Thus $\mathcal{K}_{k,j}(A, W, r_0) = span\{W, r_0, r_1, \cdots, r_{j-1}\}$ and the residuals r_i's are orthogonal to each other.*

Proof. Let $x_j = x_0 + W\hat{\xi}_j + V_j\hat{\eta}_j$, for some $\hat{\xi}_j, \hat{\eta}_j$. And we know $B = A - AW(W^TAW)^{-1}W^TA$ and $BV_j = V_jT_j + \sigma_{j+1}v_{j+1}e_j^T$, then we get

$$AV_j = AW\Delta_j + V_jT_j + \sigma_{j+1}v_{j+1}e_j^T, \; \Delta_j = (W^TAW)^{-1}W^TA.$$

Hence, $r_j = r_0 - AW\hat{\xi}_j - AV_j\hat{\eta}_j = r_0 - AW\hat{\xi}_j - (AW\Delta_j + V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)\hat{\eta}_j$.

$$\begin{aligned}
&W^Tr_j = W^Tr_0 - (W^TAW)(\hat{\xi}_j - \Delta_j\hat{\eta}_j) - W^T(V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)\hat{\eta}_j \\
&\implies 0 = -(W^TAW)\hat{\xi}_j - (W^TAW)\Delta_j\hat{\eta}_j \\
&\implies (W^TAW)\hat{\xi}_j + (W^TAW)\Delta_j\hat{\eta}_j = 0 \implies \hat{\xi}_j = -\Delta_j\hat{\eta}_j
\end{aligned}$$

Therefore, we have

$$x_j = x_0 - W\Delta_j\hat{\eta}_j + V_j\hat{\eta}_j$$

and

$$r_j = r_0 - (V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)\hat{\eta}_j.$$

We know, $V_j^Tr_j = 0 \implies 0 = V_j^Tr_0 - T_j\hat{\eta}_j \implies \hat{\eta}_j = \|r_0\|_2T_j^{-1}e_1$. Now substituting this equation in the above equation, we get

$$\begin{aligned}
r_j &= r_0 - \|r_0\|_2V_jT_j(T_j^{-1}e_1) - \|r_0\|_2\sigma_{j+1}v_{j+1}e_j^T(T_j^{-1}e_1) \\
&= (-\|r_0\|_2\sigma_{j+1}e_j^TT_j^{-1}e_1)v_{j+1} = c_jv_{j+1}
\end{aligned}$$

□

Let $T_j = L_j D_j L_j^T$ and define $P_j \equiv [p_0\ p_1\ \cdots\ p_{j-1}] = (-W\Delta_j + V_j)L_j^{-T}\Lambda_j$, where $\Lambda_j = \text{diag}\{c_0, c_1, \cdots, c_{j-1}\}$.

Proposition 13. *The approximate solution x_j, the residual r_j and the descent direction p_j satisfy the following recurrence relations:*

$$x_j = x_{j-1} + \alpha_{j-1}p_{j-1},\ r_j = r_{j-1} - \alpha_{j-1}Ap_{j-1},\ p_j = r_j + \beta_{j-1}p_{j-1} - W\hat{\mu}_j,$$

for some $\alpha_{j-1}, \beta_{j-1}, \hat{\mu}_j$. Thus $\mathcal{K}_{k,j}(A, W, r_0) = span\{W, p_0, p_1, \cdots, p_{j-1}\}$.

Proof. Let $\hat{\zeta}_j = \|r_0\|_2(L_jD_j\Lambda_j)^{-1}e_1$. We know,

$$\begin{aligned} x_j &= x_0 + \|r_0\|_2(-W\Delta_j + V_j)T_j^{-1}e_1 \\ &= x_0 + [(-W\Delta_j + V_j)L_j^{-T}\Lambda_j][\|r_0\|_2\Lambda_j^{-1}L_j^T(L_jD_jL_j^T)_j^{-1}e_1] \\ &= x_0 + P_j\zeta_j \end{aligned}$$

Now notice that $L_jD_j\Lambda_j$ is lower bidiagonal matrix, then we have,

$$\zeta_j = \begin{bmatrix} \zeta_{j-1} \\ \alpha_{j-1} \end{bmatrix}, \text{ for some scalar } \alpha_{j-1}.$$

Hence,

$$x_j = x_{j-1} + \alpha_{j-1}p_{j-1} \text{ and } r_j = r_{j-1} - \alpha_{j-1}Ap_{j-1}.$$

We know that $P_j = (-W\Delta_j + V_j)L_j^{-T}\Lambda_j$, then

$$P_j\Lambda_j^{-1}L_j^T\Lambda_j = (-W\Delta_j\Lambda_j + V_j\Lambda_j).$$

Now notice that $\Lambda_j^{-1}L_j^T\Lambda_j$ is a unit upper bidiagonal matrix. Comparing the last columns of the both sides of this equation

$$p_{j-1} - \beta_{j-2}p_{j-2} = -W\hat{\mu}_{j-1} + c_{j-1}v_j,$$

where $\hat{\mu}_{j-1} = c_{j-1}\hat{v}_j$ *and* $\beta_{j-2} = -c_{j-1}u_{j-1,j}/c_{j-2}$. □

Proposition 14. *The vectors p_j are A-orthogonal to each other, i.e, $P_j^TAP_j$ is diagonal. In addition, they are also A-orthogonal to all w_i's, i.e, $W^TAP_j = 0$.*

Proof.

$$\begin{aligned} P_j^TAP_j &= P_j^T(-AW\Delta_j + AV_j)L_j^{-T}\Lambda_j \\ &= P_j^T(V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)L_j^{-T}\Lambda_j \\ &= \Lambda_jL_j^{-1}(-\Delta_j^TW^T + V_j^T)(V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)L_j^{-T}\Lambda_j \\ &= \Lambda_jL_j^{-1}T_jL_j^{-T}\Lambda_j = \Lambda_j^2D_j, \text{ which is diagonal.} \end{aligned}$$

Consider,

$$W^TAP_j = W^T(V_jT_j + \sigma_{j+1}v_{j+1}e_j^T)L_j^{-T}\Lambda_j = 0.$$

□

Proposition 15. *The coefficients in Delfated-CG satisfy the relations*

$$\alpha_j = \frac{(r_j, r_j)}{(Ap_j, p_j)}, \quad \hat{\mu}_j = (W^TAW)^{-1}Ar_j, \quad \beta_j = \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}.$$

Proof. From the orthogonality constraints on residuals, $r_{j-1}^T r_j = 0$. Then,

$$\begin{aligned}
&\implies r_{j-1}^T(r_{j-1} - \alpha_{j-1}Ap_{j-1}) = 0\\
&\implies \alpha_{j-1} = \frac{(r_{j-1}, r_{j-1})}{(r_{j-1}, Ap_{j-1})}\\
\text{But, } &(p_{j-2}, Ap_{j-1}) = 0 \textit{ and } (W\hat{\mu}_{j-1}, Ap_{j-1}) = 0\\
&\implies \alpha_{j-1} = \frac{(r_{j-1}, r_{j-1})}{(\beta_{j-2}p_{j-2} + r_{j-1} - W\hat{\mu}_{j-1}, Ap_{j-1})}\\
&\implies \alpha_{j-1} = \frac{(r_{j-1}, r_{j-1})}{(p_{j-1}, Ap_{j-1})}
\end{aligned}$$

We know,

$$\begin{aligned}
p_j &= r_j + \beta_{j-1}p_{j-1} - W\hat{\mu}_j\\
\implies W^T Ap_j &= W^T Ar_j + \beta_{j-1}W^T Ap_{j-1} - W^T AW\hat{\mu}_j\\
\implies \hat{\mu}_j &= (W^T AW)^{-1}Ar_j\\
\text{Similarly, } p_{j-1}^T Ap_j &= p_{j-1}^T Ar_j + \beta_{j-1}p_{j-1}^T Ap_{j-1} - p_{j-1}^T AW\hat{\mu}_j\\
\implies \beta_{j-1} &= -\frac{(Ap_{j-1}, r_j)}{(p_{j-1}, Ap_{j-1})} = \frac{1}{\alpha_{j-1}}\frac{(r_j - r_{j-1}, r_j)}{(p_{j-1}, Ap_{j-1})}\\
\implies \beta_{j-1} &= \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}
\end{aligned}$$

□

Remark. *To guarantee that initial guess x_0 satisfies $W^T r_0 = 0$, we can choose x_0 in the form*

$$x_0 = x_{-1} + W(W^T AW)^{-1}W^T r_{-1},$$

where x_{-1} is arbitrary and $r_{-1} = b - Ax_{-1}$.
Then,

$$W^T r_0 = W^T r_{-1} - (W^T AW)(W^T AW)^{-1}W^T r_{-1} = 0.$$

Putting these relation together gives the following algorithm.

Algorithm 12 Deflated-CG

Choose k linearly independent vectors $w_1, w_2, \cdots, w_k$ and
    define $W = [w_1\ w_2\ \cdots\ w_k]$
Choose an initial guess x_0 such that $W^T r_0 = 0$, where $r_0 = b - Ax_0$
Solve $W^T AW\hat{\mu}_0 = W^T Ar_0$ for $\hat{\mu}_0$ and set $p_0 = r_0 - W\hat{\mu}_0$
for $j = 1, 2, \cdots, m$, **do**
    $\alpha_{j-1} = (r_{j-1}, r_{j-1})/(Ap_{j-1}, p_{j-1})$
    $x_j = x_{j-1} + \alpha_{j-1}p_{j-1}$
    $r_j = r_{j-1} - \alpha_{j-1}Ap_{j-1}$
    $\beta_{j-1} = (r_j, r_j)/(r_{j-1}, r_{j-1})$
    Solve $W^T AW\hat{\mu}_j = W^T Ar_j$ for $\hat{\mu}_j$
    $p_j = r_j + \beta_{j-1}p_{j-1} - W\hat{\mu}_j$
EndDo

References

[1] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS publishing company, Boston, MA, 1996.

[2] Y. Saad, M. Yeung, J. Erhel, F. Guyomarc'H, A deflated version of the conjugate gradient algorithm, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[3] W. D. Joubert and T. A. Manteuffel, Iterative Methods for Nonsymmetric Linear Systems, Academic Press, New York, 1990, pp. 149–171