Iterative Methods

Jayadev Naram

October 6, 2019

Contents

1 General Projection Methods **2**

2 One-Dimensional Projection Methods **5**
2.1 Steepest Descent 5

3 Krylov Subspace Methods **7**

4 Arnoldi's Method for Linear Systems (FOM) **8**
4.1 Variation 1: Restarted FOM 10
4.2 Variation 1: IOM and DIOM 10

5 Symmetric Lanczos Algorithm **12**

6 Conjugate Gradient **13**

1 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 $\mathcal{K}$ *and* $\mathcal{L}$
3: Choose basis $V{=}[v_1, \cdots, v_m]$, $W{=}[w_1, \cdots, w_m]$ $for\ \mathcal{K}\ and\ \mathcal{L}$
4: $r \leftarrow b - Ax$
5: $y \leftarrow (W^T AV)^{-1}W^T r$
6: $x \leftarrow x + Vy$
7: **until** Convergence

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

Proposition. *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 1. *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 1.1. *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. *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. *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

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

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

Theorem 2. *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

$$\begin{aligned} &\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(b - A\tilde{x}) = 0, \; \textit{where} \; \tilde{x} = x_0 + \delta, \; \delta \in \mathcal{K}. \\ &\implies \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(b - Ax_0 - A\delta) = 0 \\ &\implies \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} r_0 = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\delta \end{aligned}$$

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

$$\begin{aligned} &\implies r_0 = A\delta, \; \text{(since } r_0 \in \mathcal{K} \text{ and } \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\delta = A\delta) \\ &\implies A\tilde{x} = b \end{aligned}$$

$\square$

Theorem 3 (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 $b \in \mathcal{K}$,

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

□

2 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,

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

2.1 Steepest Descent

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

Lemma 4 (**Kantorovich inequality**). *Let B be any real SPD matrix and* $\lambda_1,\ \lambda_n$ *its largest and smallest eigenvalues. Then,*

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

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, $B = Q^T D Q$.

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

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 $\lambda = \frac{\lambda_1 + \lambda_n}{2}$ yielding,

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

$\square$

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 5. *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,

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

From Kantorovich inequality,

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

□

3 Krylov Subspace Methods

We define Krylov Subspace to be

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

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 6. *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,

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

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

4 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. If h_{j+1,j} = 0 Stop
9:     v_{j+1} = w_j / h_{j+1,j}
10: EndDo
```

Proposition. *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)

1: Compute $r_0 = b - Ax_0,\ \beta = \|r_0\|_2,\ and\ v_1 = r_0/\beta$
2: Define the $m \times m$ matrix $H_m = \{h_{ij}\}_{i,j=1,2,\cdots,m}; Set\ H_m = 0$
3: **for** $j = 1, 2, \cdots, m$ **do**
4: Compute $w_j = Av_j$
5: **for** $i = 1, 2, \cdots, j$ **do**
6: $h_{ij} = (w_j, v_i)$
7: $w_j = w_j - h_{ij}v_i$
8: EndDo
9: $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ Stop
10: $v_{j+1} = w_j/h_{j+1,j}$
11: EndDo
12: Compute $y_m = H_m^{-1}\beta e_1$ and $x_m = x_0 + V_m y_m$

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

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

and, therefore,

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

Proof.

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

□

4.1 Variation 1: Restarted FOM

Algorithm 5 Restarted FOM (FOM(m))

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

4.2 Variation 1: IOM and DIOM

Algorithm 6 Incomplete Orthogonalization Method (IOM)

1: Compute $r_0 = b - Ax_0,\ \beta = \|r_0\|_2,\ and\ v_1 = r_0/\beta$
2: Define the $m \times m$ matrix $H_m = \{h_{ij}\}_{i,j=1,2,\cdots,m}; Set\ H_m = 0$
3: **for** $j = 1, 2, \cdots, m$ **do**
4: Compute $w_j = Av_j$
5: **for** $i = max\{1, j-(k-1)\}, 2, \cdots, j$ **do**
6: $h_{ij} = (w_j, v_i)$
7: $w_j = w_j - h_{ij}v_i$
8: EndDo
9: $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ Stop
10: $v_{j+1} = w_j/h_{j+1,j}$
11: EndDo
12: Compute $y_m = H_m^{-1}\beta e_1$ and $x_m = x_0 + V_m y_m$

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

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

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. *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 $r_m \perp \mathcal{L}_m$. □

5 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

1: Compute $r_0 = b - Ax_0,\ \beta = \|r_0\|_2,\ and\ v_1 = r_0/\beta$
2: **for** $j = 1, 2, \cdots, m$ **do** ▷ Orthogonalization Procedure
3: $\quad w_j = Av_j - \beta_j v_{j-1}$
4: $\quad \alpha_j = (w_j, v_j)$
5: $\quad w_j = w_j - \alpha_j v_j$
6: $\quad \beta_{j+1} = \|w_j\|_2$. If $\beta_{j+1} = 0$ then Stop
7: $\quad v_{j+1} = w_j/\beta_{j+1}$
8: EndDo
9: Set $T_m = tridiag(\beta_i, \alpha_i, \beta_{i+1})$, and $V_m = [v_1, \cdots, v_m]$.
10: Compute $y_m = H_m^{-1}\beta e_1$ and $x_m = x_0 + V_m y_m$

6 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