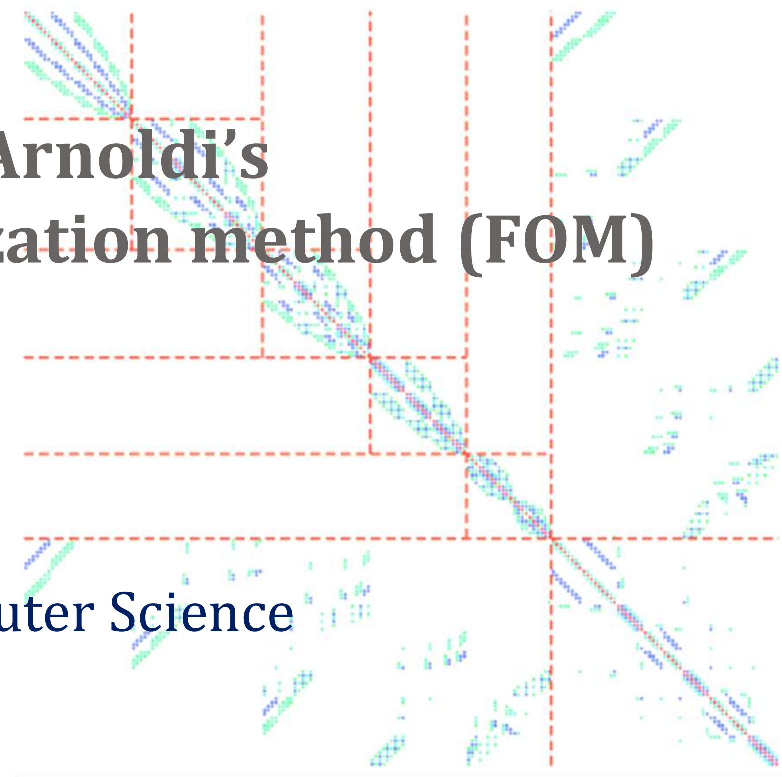
Numerical Methods of Linear Algebra for Sparse Matrices

Lecture 13

Krylov subspace methods based on Arnoldi's orthogonalization. Full Orthogonalization method (FOM)



Anna Nasedkina

Department of Mathematical Modeling

Institute of Mathematics, Mechanics and Computer Science

Southern Federal University

Krylov Subspace methods based on Arnoldi's orthogonalization

- Definition of Krylov subspace and Krylov subspace projection method
- Arnoldi's method of orthogonalization
- Full Orthogonalization method (FOM)

Multi-dimensional projection method

For the linear system $Ax = b$, $A \in \mathbb{C}^{n \times n}$, $b \in \mathbb{C}^n$
consider the case when $m > 1$ and $m \ll n$.
Search the approximate solution $x_m \in x_0 + K_m$, where

- x_0 is the initial guess, $r_0 = b - Ax$ is the initial residual
- K_m is the search subspace, $\dim(K_m) = m$
- L_m is the subspace of constraints, $\dim(L_m) = m$
- $r_m = b - Ax_m \perp L_m$ is the residual

Approximate solution $x_m = x_0 + \delta_m$, $\delta_m \in K_m$ is correction vector.
$r_m = r_0 - A\delta_m \perp L_m$
How to choose δ_m? Ideally it should give zero residual:
$r_m = r_0 - A\delta_m = 0$. Hence $\delta_m^{best} = A^{-1} r_0$
Computing δ_m^{best} from solving the system $A\delta_m = r_0$ is not efficient.
In projection method $\delta_m = V_m y$, where $y \in \mathbb{C}^m$, $V_m \in \mathbb{C}^{n \times m}$
$\delta_m \approx A^{-1} r_0$

Krylov subspace projection method

Take $K_m = K_m(A, r_0)$, where K_m is Krylov subspace

Krylov subspace of dimension m, induced by the matrix A and the vector v, is defined as $K_m(A, v) = span\{v, Av, A^2v, \dots, A^{m-1}v\}$

In Krylov subspace methods, the search subspace is the Krylov subspace, induced by the matrix A and the initial residual vector r_0: $K_m = K_m(A, r_0)$

Any Krylov subspace method implicitly generates a polynomial $q_{m-1}(A)$ of a degree $m - 1$:
$q_{m-1}(A)r_0 = \alpha_0 r_0 + \alpha_1 A r_0 + \alpha_2 A^2 r_0 + \cdots + \alpha_{m-1} A^{m-1} r_0$

Approximate solution
$x_m = x_0 + q_{m-1}(A)r_0 \approx A^{-1}b$

Arnoldi method of orthogonalization

How to build Krylov subspace $K_m = K_m(A, v_1)$? The subspace is defined by its basis.

The basis for Krylov subspace can be constructed by Arnoldi process.

Arnoldi process is the Gram-Schmidt process which builds *orthonormal* set of vectors $\{v_1, v_2, ..., v_m\} = V_m$ for a given vector v_1 and given matrix A that is the basis of $K_m(A, v_1)$

Recall the definition of orthonormal set $\{v_1, v_2, ..., v_m\}$: $(v_i, v_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$; $\|v_i\|_2 = 1,\ i = \overline{1, m}$

Basic algorithm of Arnoldi orthogonalization

1) Choose v_1: $\|v_1\|_2 = 1$. Matrix A and dimension of subspace m are given.

2) Loop for j from 1 to m.

2.1) $h_{ij} := (Av_j, v_i),\ i = \overline{1, j}$

2.2) $w_j := Av_j - \sum_{i=1}^{j} h_{ij} v_i$

2.3) $h_{j+1,j} := \|w_j\|_2$. If $h_{j+1,j} = 0 \Rightarrow$ stop, else

2.4) $v_{j+1} := \dfrac{w_j}{h_{j+1,j}}$

Arnoldi process for numerical implementation

- Arnoldi process is the modified Gram-Schmidt process applied to Krylov subspace $K_m(A, v_1)$ induced by matrix A and vector v_1

1) Choose $v_1 : \ \|v_1\|_2 = 1$. Matrix A and dimension of subspace m are given.

2) Loop for j from 1 to m.

 2.1) $w_j = Av_j$

 2.2) Loop for i from 1 to j $h_{ij} := (w_j, v_i),\ w_j := w_j - h_{ij} v_i$

 2.3) $h_{j+1,j} := \|w_j\|_2$. If $h_{j+1,j} = 0 \Rightarrow$ stop, else

 2.4) $v_{j+1} := \dfrac{w_j}{h_{j+1,j}}$

Properties of Arnoldi process

Property 1. If all m steps of Arnoldi process are complited, then the set of vectors $\{v_1, v_2, \ldots, v_m\}$ is the orthonormalized basis of Krylov subspace $K_m(A, v_1)$.

Property 2. From Arnoldi algorithm form the matrices:

$$V_m = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \ldots & v_m \\ | & | & & | \end{pmatrix} \in \mathbb{C}^{n\times m};\; V_{m+1} = \begin{pmatrix} | & | & & | & | \\ v_1 & v_2 & \ldots & v_m & v_{m+1} \\ | & | & & | & | \end{pmatrix} \in \mathbb{C}^{n\times(m+1)}$$

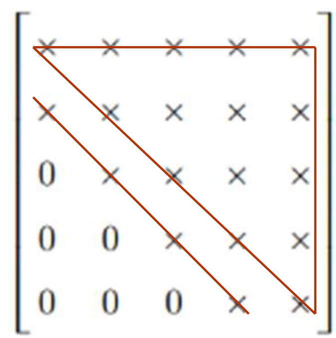
$$\overline{H}_m = \{\tilde{h}_{ij}\},\; \tilde{h}_{ij} = \begin{cases} 0, & i > j+1 \\ h_{ij}, & i \le j+1 \end{cases};\; H_m = \begin{pmatrix} \tilde{h}_{11} & \ldots & \tilde{h}_{1m} \\ \ldots & & \ldots \\ \tilde{h}_{m1} & \ldots & \tilde{h}_{mm} \end{pmatrix};\; H_m \in \mathbb{C}^{m\times m}$$

$\overline{H}_m \in \mathbb{C}^{(m+1)\times m}$ is Hessenberg matrix of the size $m+1$ by m

Then the following Arnoldi relation holds:

1) $AV_m = V_m H_m + w_m e_m^T = V_{m+1} \overline{H}_m$

2) $V_m^H A V_m = H_m$



Upper Hessenberg

Arnoldi relation 1

1) $AV_m = V_m H_m + w_m e_m^T = V_{m+1} \overline{H}_m$

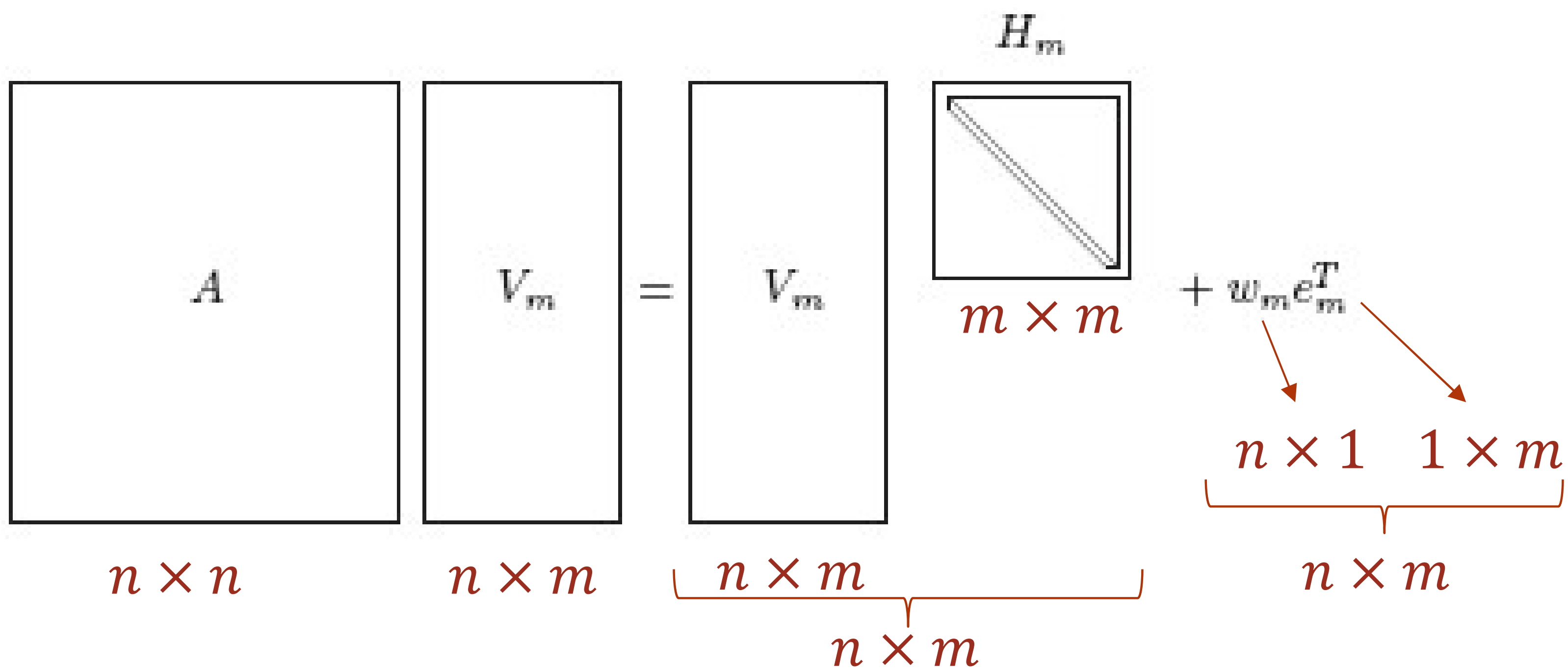
FOM GMRES

Recall that at the last step of Arnoldi algorithm when $j = m$: $v_{m+1} = \dfrac{w_m}{h_{m+1,m}}$

From (1) we can derive two methods:

a) FOM: $AV_m = V_m H_m + w_m e_m^T = V_m H_m + v_{m+1}\, h_{m+1,m}\, e_m^T$

b) GMRES: $AV_m = V_{m+1} \overline{H}_m$



Arnoldi relation 2

2) $V_m^H A\, V_m = H_m$; $V_{m+1}^H A\, V_m = \overline{H}_m$

V_m is unitary non-square matrix (orthonormal columns):
$V_m^H V_m = I_m$

(2) shows that A can be transformed to H_m: $A \sim H_m$

$$\underset{m\times m}{H_m} = \underset{m\times n}{V_m^H} * \underset{n\times n}{A} * \underset{n\times m}{V_m}$$

$$\underset{(m+1)\times m}{\overline{H}_m} = \underset{(m+1)\times n}{V_{m+1}^H} * \underset{n\times n}{A} * \underset{n\times m}{V_m}$$

Arnoldi process for solving the linear system Ax=b

- When applied to solving the system $Ax = b$, Arnoldi process constructs an orthonormal basis for the search subspace $K_m = K_m(A, r)$, where K_m is the Krylov subspace and $r = b - Ax$ is the residual vector

Arnoldi algorithm

1. define zero matrix $\underline{H}_m \in \mathbb{R}^{(m+1)\times m}$ and zero matrix $V_{m+1} \in \mathbb{R}^{n\times(m+1)}$
2. $\beta := \|r\|_2$
 $v_1 := r/\beta$
3. for $j = 1, 2, \ldots, m$ do

   every loop cycle gives $v_{j+1} =$ `V(:,j+1)` and $(h_{1,j}, \ldots, h_{j+1,j})^T =$ `H(1:j+1,j)`

4. $\quad w_j := Av_j$
5. $\quad$ for $i = 1, 2, \ldots, j$ do
6. $\qquad h_{ij} := (w_j, v_i)$
7. $\qquad w_j := w_j - h_{ij} v_i$
8. $\quad$ endfor
9. $\quad h_{j+1,j} := \|w_j\|_2$.
 $\quad$ If $h_{j+1,j} = 0$ stop.
10. $\quad v_{j+1} := w_j / h_{j+1,j}$
11. endfor

Arnoldi in Matlab

```
H=zeros(m+1,m);
V=zeros(n,m+1);
beta=norm(r);
V(:,1)=(1/beta)*r;
for j=1:m,
   wj=A*V(:,j);
   for i=1:j,
      H(i,j)=(V(:,i))'*wj;
      wj=wj-H(i,j)*V(:,i);
   end
   H(j+1,j)=norm(wj);
   if(H(j+1,j)==0), break, end
V(:,j+1)=(1/H(j+1,j))*wj;
end
```

Arnoldi process for solving the linear system Ax=b

- Classical examples of methods based on Arnold orthogonalization process are FOM (Full Orthogonalization Method) and GMRES (Generalized Minimal Residual)

- In these methods, the solution is searched in the form $x_m = x_0 + \delta_m$, $\delta_m \in K_m(A, r_0)$ is correction vector,

  $K_m(A, r_0) = span\{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\}$ is Krylov subspace

- Orthogonal basis of $K_m(A, r_0)$ is produced by Arnoldi process, where the basis vectors are the columns of V_m:

  $\delta_m \in K_m(A, r_0) \Leftrightarrow \exists y \in C^m\!: \delta_m = V_m y$

- FOM uses the first part of Arnoldi relation: $AV_m = V_m H_m + w_m e_m^T$ and solves the linear system of the size m with matrix H_m

- GMRES uses the second part of Arnoldi relation: $AV_m = V_{m+1} \overline{H}_m$ and solves the linear system of size the $m + 1$ by m with matrix $\overline{H}_m$

Derivation of Full Orthogonalization method (FOM)

Consider projection method with Krylov subspace as the search subspace.

Search subspace is Krylov subspace $K_m(A, r_0) = span\{r_0, Ar_0, A^2 r_0, ..., A^{m-1} r_0\}$

Subspace of constraints is $L_m = K_m = K_m(A, r_0)$

$$V_m = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_m \\ | & | & & | \end{pmatrix}$$ is the basis in K_m, $W_m = V_m$ is the basis in $L_m = K_m$

Approximate solution is given by $x_m = x_0 + \delta_m = x_0 + V_m y_m;\ \delta_m \in K_m(A, r_0)$

y_m can be found from solving the system $(W_m{}^H A V_m) y = W_m{}^H r_0 \Rightarrow$

formally $y_m = (W_m{}^H A V_m)^{-1} W_m{}^H r_0$. When $W_m = V_m$ we have:

$y_m = (V_m{}^H A V_m)^{-1} V_m{}^H r_0 = H_m^{-1} V_m^H r_0$ from Arnoldi relation

Derivation of Full Orthogonalization method (FOM)

We build orthonormal basis in Arnoldi process: $\|v_1\|_2 = 1$

As we take Krylov subspace induced by initial residual, we can take

the first vector as normalized initial residual: $v_1 = \dfrac{r_0}{\|r_0\|_2}$

Denote $\beta = \|r_0\|_2$, then $r_0 = \beta v_1$

Hence $y_m = H_m^{-1} V_m^H r_0 = H_m^{-1} V_m^H \beta v_1 = H_m^{-1} \beta e_1$, because

$$V_m^H \beta v_1 = \beta V_m^H v_1 = \beta \begin{pmatrix} -v_1- \\ -v_2- \\ \dots \\ -v_m- \end{pmatrix} \begin{pmatrix} | \\ v_1 \\ | \end{pmatrix} = \beta \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \beta e_1 \text{ (e_1 is the unit ort)}$$

So $y_m = H_m^{-1} \beta e_1$, and we find y_m from solving the system

$$H_m y = \beta e_1$$

Here $H_m \in \mathbb{C}^{m \times m}$, $y \in \mathbb{C}^m$, $e_1 \in \mathbb{R}^m$ is the unit ort

We have small linear system with m equations and m unknowns, which is easier to solve than initial system $Ax = b$ with large matrix A.

Small system in FOM

We find y from the system $H_m y = \beta e_1$ with m equations and m unknowns

$$y^T = y_m{}^T = \left(y_m^1, \quad y_m^2, \quad \dots \quad y_m^m \right),$$

where $y_m^1, \quad y_m^2, \quad \dots \quad y_m^m$ are the unknowns to be found from the system

$$\begin{cases} h_{11} y_m^1 + h_{12} y_m^2 + \quad \dots \quad + h_{1m} y_m^m = \beta \\ h_{21} y_m^1 + h_{22} y_m^2 + \quad \dots \quad + h_{2m} y_m^m = 0 \\ \dots \\ h_{m,m-1} y_m^{m-1} + h_{mm} y_m^m = 0 \end{cases}$$

For example for $m = 4$ the matrix form of the system is:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ 0 & h_{32} & h_{33} & h_{34} \\ 0 & 0 & h_{43} & h_{44} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} \beta \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Full Orthogonalization method (FOM)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$,

calculate residual norm $\beta = \|r_0\|_2$, and the first vector $v_1 = \dfrac{r_0}{\beta}$

2. Define $H_m = \{h_{ij}\} \in \mathbb{C}^{m\times m}$, $V_m \in \mathbb{C}^{n\times m}$. Set $H_m = 0$

3. Build the basis for Krylov subspace using Arnoldi process:

Loop for j from 1 to m.

3.1) $w_j = Av_j$

3.2) Loop for i from 1 to j $h_{ij} := (w_j, v_i)$, $w_j := w_j - h_{ij}v_i$

3.3) $h_{j+1,j} := \|w_j\|_2$. If $h_{j+1,j}=0 \Rightarrow m := j$, go to step 4, else

3.4) $v_{j+1} := \dfrac{w_j}{h_{j+1,j}}$

4. Find y_m from solving the system $H_m y = \beta e_1$

5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

The residual in FOM can be calculated **without** general formula $r_m = b - Ax_m$

Calculating residual in FOM

FOM depends on the dimension of Krylov subspace m.

It is possible to select m so, that $\|r_m\|_2 < \varepsilon$

In fact, the residual can be computed inexpensively, without having to compute approximate solution x_m

Proposition. The residual vector r_m of the approximate solution x_m computed by FOM algorithm is defined as

$r_m = -h_{m+1,m} e_m^T y_m v_{m+1}$ and residual norm as $\|r_m\|_2 = h_{m+1,m} \left| e_m^T y_m \right|$

△ Proove the formula for residual norm: 2) $\|r_m\|_2 = h_{m+1,m} \left| e_m^T y_m \right|$

As from 1) $r_m = -h_{m+1,m} e_m^T y_m v_{m+1} \Rightarrow$

$$\|r_m\|_2 = \left\| -h_{m+1,m} e_m^T y_m v_{m+1} \right\|_2 = h_{m+1,m} \left| e_m^T y_m \right| \|v_{m+1}\|_2 = h_{m+1,m} \left| e_m^T y_m \right|$$

because $h_{m+1,m}$ is computed as a norm, so $h_{m+1,m} > 0$,

v_{m+1} is the basis vector from orthonormal basis, so $\|v_{m+1}\|_2 = 1$, $e_m^T y_m$ is a scalar. □

Calculating residual in FOM

▵ Proove the formula for residual: 1) $r_m = -h_{m+1,m} e_m^T y_m v_{m+1}$

Recall that $r_m = b - Ax_m = b - A(x_0 + V_m y_m) = b - Ax_0 - AV_m y_m =$

$= r_0 - AV_m y_m$

From Arnoldi relation $AV_m = V_m H_m + w_m e_m^T = V_m H_m + v_{m+1} h_{m+1,m} e_m^T$,

from FOM algorithm $r_0 = \beta v_1 \Rightarrow$

$r_m = r_0 - AV_m y_m = \beta v_1 - (V_m H_m + v_{m+1} h_{m+1,m} e_m^T) y_m = \beta v_1 - \underline{V_m H_m y_m} - v_{m+1} h_{m+1,m} e_m^T y_m.$

From Arnoldi algorithm $y_m = H_m^{-1} \beta e_1 \Rightarrow$

$r_m = \beta v_1 - \underline{V_m H_m H_m^{-1} \beta e_1} - v_{m+1} h_{m+1,m} e_m^T y_m = \beta v_1 - \underline{\underline{V_m \beta e_1}} - v_{m+1} h_{m+1,m} e_m^T y_m$

$$V_m \beta e_1 = \beta V_m e_1 = \beta \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \beta v_1 \Rightarrow$$

$r_m = \beta v_1 - V_m \beta e_1 - v_{m+1} h_{m+1,m} e_m^T y_m = \beta v_1 - \beta v_1 - v_{m+1} h_{m+1,m} e_m^T y_m =$

$= -v_{m+1} h_{m+1,m} e_m^T y_m \Rightarrow r_m = -h_{m+1,m} e_m^T y_m v_{m+1}$ □

Modifications of FOM

A step of FOM costs approximately $2Nz(A)+2mn$, where $Nz(A)$ is the number of nonzero entries in A.

Together with storing the basis V_m, Hessenberg matrix H_m, additional vectors for the current solution and right-hand side, the total is roughly $(m+3)n+\frac{m^2}{2}$

As m increases, the computational cost increases as $O(m^2 n)$. The memory cost increases as $O(mn)$.

The idea of Krylov subspace methods is to use $m << n$.

So in order not to increase m to the value of n, there are two variations of FOM:

1) restart FOM for the fixed value of m

2) truncate the orthogonalization in Arnoldi process (see details in Yosef Saad book, p. 168)

Restarted FOM(m)

1. Take initial guess x_0, calculate initial residual $r_0 = b - Ax_0$,

calculate residual norm $\beta = \|r_0\|_2$, and the first vector $v_1 = \frac{r_0}{\beta}$

2. Define $H_m = \{h_{ij}\} \in \mathbb{C}^{m \times m}$, $V_m \in \mathbb{C}^{n \times m}$. Set $H_m = 0$

3. Build the basis for Krylov subspace using Arnoldi process.

4. Find y_m from solving the system $H_m y = \beta e_1$

5. Calculate residual r_m, estimate residual norm: $\|r_m\|_2 < \varepsilon$.

6. Calculate new approximate solution $x_m = x_0 + V_m y_m$

If satisfied, then stop. Else

7. Set $x_0 := x_m$ and go to step 1.

- Convergence of restarted FOM: sometimes small m is sufficient, sometimes the largest possible m is necessary
- Variation of restarted FOM: start with $m = 1$ and for every run of the algorithm increment m by 1 until certain m_{max} is reached