

Krylov Subspace Iteration Methods

Anastasia Filimon

ETH Zurich

29 May 2008

- The methods represent iterative techniques for solving large linear systems

$$Ax = b,$$

where A is non-singular $n \times n$ -matrix, b is n -vector, n is large.

- They are based on projection processes onto Krylov subspaces.

- Krylov subspace generated by an $n \times n$ -matrix A , and an n -vector b is the subspace spanned by the vectors of the Krylov sequence:

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

- The projection method seeks an approximate solution x_m from an affine subspace $x_0 + K_m$ by imposing condition $b - Ax_m \perp L_m$, L_m is another subspace of dimension m , x_0 is an initial guess to the solution. In the case of Krylov subspace methods $K_m = K_m(A, r_0)$, $r_0 = b - Ax_0$ is an n -vector

$$K_m = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}.$$

- **Property 1.** K_m is the subspace of all vectors in R^n which can be written as

$$x = p(A)v,$$

where p is a polynomial of degree not exceeding $m - 1$.

- The different versions of Krylov subspace methods arise from different choices of the subspace L_m and from the ways in which the system is preconditioned. Two broad choices for L_m give rise to the best-known techniques:

$$L_m = K_m \quad \text{FOM}$$

$$L_m = AK_m \quad \text{GMRES, MINRES.}$$

Arnoldi Modified Gram-Schmidt Method

- This is an algorithm for building an orthogonal basis of the Krylov subspace K_m .

ALGORITHM Arnoldi-Modified Gram-Schmidt

1. Choose a vector v_1 of norm 1
2. For $j = 1, 2, \dots, m$ Do:
3. Compute $w_j := Av_j$
4. For $i = 1, \dots, j$ 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

- At each step, the algorithm multiplies Arnoldi vector v_j by A and then orthonormalizes the resulting vector w_j against all previous v_j 's by a standard Gram-Schmidt procedure.

- **Proposition 1.** Assume the Arnoldi Algorithm does not stop before the m -th step. Then the vectors v_1, v_2, \dots, v_m form an orthonormal basis of the Krylov subspace

$$K_m = \text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}.$$

- **Proposition 2.** Projection method onto the subspace K_j will be exact when a breakdown occurs at step j .
- **Proposition 3.** Denote by V_m the $n \times m$ matrix with column vectors v_1, \dots, v_m , by \overline{H}_m , the $(m+1) \times m$ Hessenberg matrix whose nonzero entries h_{ij} are defined by Arnoldi Modified Gram-Schmidt Algorithm and by H_m the matrix obtained from \overline{H}_m by deleting its last row. Then the following relations hold:

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

$$V_m^T AV_m = H_m.$$

Full Orthogonalization Method (FOM)

- Given the initial guess x_0 to the original linear system $Ax = b$, consider the projection method described before, which takes $L_m = K_m(A, r_0)$, where $r_0 = b - Ax_0$. If $v_1 = \frac{r_0}{\|r_0\|_2}$ in Arnoldi's method and set $\beta = \|r_0\|_2$, then

$$\begin{aligned}V_m^T A V_m &= H_m \quad \text{from Proposition 3, and} \\V_m^T r_0 &= V_m^T (\beta v_1) = \beta e_1.\end{aligned}$$

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

$$x_m = x_0 + V_m y_m, \quad y_m = H_m^{-1}(\beta e_1)$$

Full Orthogonalization Method (FOM)

- The presented algorithm depends on a parameter m which is the dimension of the Krylov subspace. In practice it is desirable to select m in a dynamic fashion. This would be possible if the residual norm of x_m is available without computation x_m itself.
- **Proposition 4.** The residual vector of the approximate solution x_m computed by the FOM Algorithm is such that

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

ALGORITHM 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 $H_m = \{h_{ij}\}_{i,j=1,\dots,m}$; Set $H_m = 0$
3. For $j = 1, 2, \dots, m$ Do:
4. Compute $w_j := Av_j$
5. For $i = 1, \dots, j$ Do:
6. $h_{ij} = (w_j, v_i)$
7. $w_j := w_j - h_{ij}v_i$
8. EndDo
9. Compute $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ set $m := j$ and Goto 12
10. Compute $v_{j+1} = w_j/h_{j+1,j}$.
11. EndDo
12. Compute $y_m = H_m^{-1}(\beta e_1)$ and $x_m = x_0 + V_m y_m$

Generalized Minimum Residual Method (GMRES)

- The method is a projection method based on taking $L_m = AK_m$, in which K_m is the m -th Krylov subspace with $v_1 = r_0 / \|r_0\|_2$. Such a technique minimizes the residual norm over all vectors in $x_0 + K_m$. The implementation of an algorithm based on this approach is similar to that of the FOM algorithm.
- Any vector x in $x_0 + K_m$ can be written as $x = x_0 + V_m y$, where y is an m -vector.

- Define

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

- Using relation from Proposition 3

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) = r_0 - AV_m y = \beta v_1 - V_{m+1} \bar{H}_m y \\ &= V_{m+1} (\beta e_1 - \bar{H}_m y). \end{aligned}$$

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

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

Generalized Minimum Residual Method (GMRES)

- The GMRES approximation is the unique vector of $x_0 + K_m$ which minimizes $J(y)$, i.e.

$$\begin{aligned}x_m &= x_0 + V_m y_m, \text{ where} \\y_m &= \arg \min_y \|\beta e_1 - \bar{H}_m y\|_2\end{aligned}$$

- The minimizer is inexpensive to compute since it requires the solution of an $(m + 1) \times m$ least-squares problem where m is typically small.

ALGORITHM GMRES

1. Compute $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$, and $v_1 := r_0/\beta$
2. Define the $(m+1) \times m$ matrix $\bar{H}_m = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}$. Set $\bar{H}_m = 0$.
3. For $j = 1, 2, \dots, m$ Do:
4. Compute $w_j := Av_j$
5. For $i = 1, \dots, j$ 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 $h_{j+1,j} = 0$ set $m := j$ and go to 12
10. $v_{j+1} = w_j/h_{j+1,j}$
11. EndDo
12. Compute y_m the minimizer of $\|\beta e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$.

Generalized Minimum Residual Method (GMRES).

Practical Implementation issues

- A clear difficulty with GMRES algorithm is that it does not provide the approximate solution x_m explicitly at each step. As a result, it is not easy to determine when to stop. However, there is a solution related to the way in which the least-squares problem is solved.
- In order to solve the least-squares problem $\min \|\beta e_1 - \bar{H}_m y\|$, it is natural to transform the Hessenberg matrix into upper triangular form by using plane rotations.

Generalized Minimum Residual Method (GMRES).

Practical Implementation issues

- Define the product of matrices Ω_i

$$Q_m = \Omega_m \Omega_{m-1} \dots \Omega_1,$$

- $\bar{R}_m, \bar{g}_m = (\gamma_1, \dots, \gamma_{m+1})^T$ the resulting matrix and right-hand side

$$\bar{R}_m = \bar{H}_m^{(m)} = Q_m \bar{H}_m,$$

$$\bar{g}_m = Q_m(\beta e_1) = (\gamma_1, \dots, \gamma_{m+1})^T.$$

- Denote by R_m the $m \times m$ upper triangular matrix obtained from \bar{R}_m by deleting its last row and by g_m the m -dimensional vector obtained from \bar{g}_m by deleting its last component.

Generalized Minimum Residual Method (GMRES).

Practical Implementation issues

- Then,

1. Vector y_m which minimizes $\|\beta e_1 - \bar{H}_m y\|_2$ is given by

$$y_m = R_m^{-1} g_m.$$

2. The residual vector at step m satisfies

$b - Ax_m = V_{m+1}(\beta e_1 - \bar{H}_m y) = V_{m+1} Q_m^T (\gamma_{m+1} e_{m+1})$ and, as a result,

$$\|b - Ax_m\|_2 = |\gamma_{m+1}|.$$

- This was the process for computing the least-squares solution y_m . The process must be stopped if the residual norm $|\gamma_{m+1}|$ is small enough. The last rows of \bar{R}_m and \bar{g}_m are deleted and the resulting upper triangular system is solved to obtain y_m . Then the approximate solution $x_m = x_0 + V_m y_m$ is computed.

The Symmetric Lanczos Algorithm

- This algorithm can be viewed as a simplification of Arnoldi's method for the particular case when the matrix is symmetric. When A is symmetric, then the Hessenberg matrix H_m becomes symmetric tridiagonal.

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

ALGORITHM The Lanczos Algorithm

1. Choose an initial vector v_1 of norm unity. Set $\beta_1 \equiv 0, v_0 \equiv 0$
2. For $j = 1, 2, \dots, m$ Do:
3. $w_j := Av_j - \beta_j v_{j-1}$
4. $\alpha_j := (w_j, v_j)$
5. $w_j := w_j - \alpha_j v_j$
6. $\beta_{j+1} := \|w_j\|_2$. If $\beta_{j+1} = 0$ then Stop
7. $v_{j+1} := w_j / \beta_{j+1}$
8. EndDo

The Conjugate Gradient Algorithm

- This algorithm is one of the best known iterative techniques for solving sparse Symmetric Positive Definite linear systems.
- Assume we need to minimize the following function

$$f(x) = \frac{1}{2}x^T Ax - x^T b$$

where A is $n \times n$ -matrix positive definite and symmetric, b is n -vector.

- The minimum value of $f(x)$ is $-b^T A^{-1} b/2$, achieved by setting $x = A^{-1} b$. Therefore, minimizing $f(x)$ and solving $Ax = b$ are equivalent problems if A is symmetric positive definite.

The Conjugate Gradient Algorithm

- The vector x_{j+1} can be expressed as .

$$x_{j+1} = x_j + \alpha_j p_j.$$

- Therefore, the residual vectors must satisfy the recurrence

$$r_{j+1} = r_j - \alpha_j A p_j.$$

- To have r_j 's orthogonal it is necessary that

$$(r_j - \alpha_j A p_j, r_j) = 0$$

and, as a result,

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

The Conjugate Gradient Algorithm

- The first basis vector p_1 is the gradient of f at x_0 , which equals to $Ax_0 - b$. The other vectors in the basis will be conjugate to the gradient. Each next p_{k+1} is defined to be in the direction closest to the gradient r_k under the conjugacy constraint. This direction is given by the projection of r_k onto the space orthogonal to p_k with respect to the inner product induced by A .

$$p_{j+1} = r_{j+1} + \beta_j p_j,$$

$$(Ap_j, r_j) = (Ap_j, p_j - \beta_{j-1} p_{j-1}) = (Ap_j, p_j)$$

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

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

ALGORITHM Conjugate Gradient

1. Compute $r_0 := b - Ax_0, p_0 := r_0$.
2. For $j = 0, 1, \dots$, until convergence Do:
3. $\alpha_j := (r_j, r_j) / (Ap_j, p_j)$
4. $x_{j+1} := x_j + \alpha_j p_j$
5. $r_{j+1} := r_j - \alpha_j Ap_j$
6. $\beta_j := (r_{j+1}, r_{j+1}) / (r_j, r_j)$
7. $p_{j+1} := r_{j+1} + \beta_j p_j$
8. EndDo

- The process stops if r_{j+1} is "sufficiently small".

Convergence Analysis

- One of the main tool used in the analysis of convergence behavior is Chebyshev polynomials.
- **Lemma 1.** 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 is of the form

$$x_m = x_0 + q_m(A)r_0,$$

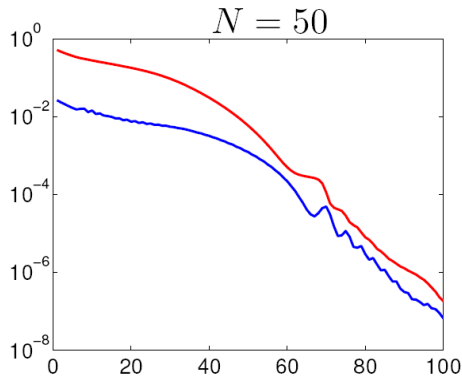
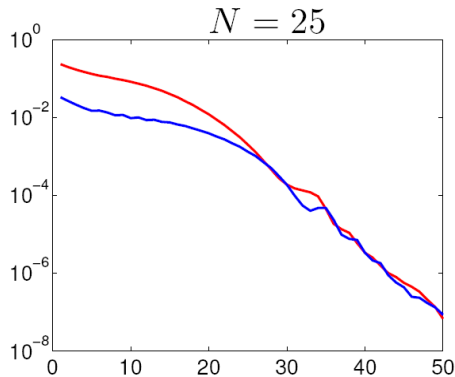
where q_m is a polynomial of degree $m - 1$ such that

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

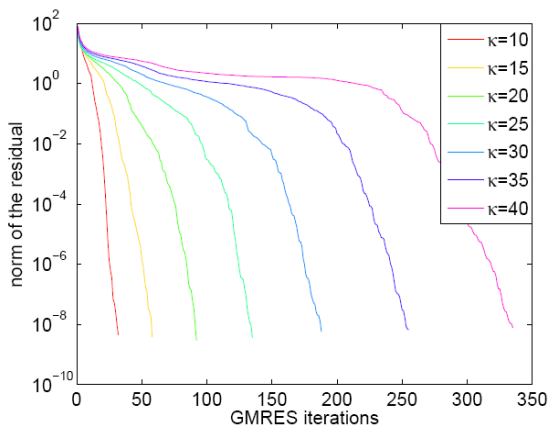
- **Theorem 2.** Let x_m be the approximate solution obtained from the m -th step of the CG algorithm, and x_* is the exact solution. Then

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

Convergence Analysis



Convergence Analysis



GMRES residual norms show a period of stagnation followed by rapid convergence. The above plots are for $\kappa=10, 15, \dots, 40$.