

# CAAM 454: Aftersnotes on Krylov-Subspace Iterative Methods

Yin Zhang (draft)

CAAM, Rice University, Houston, TX 77005

February 3, 2007

Revised January 24, 2008

## 1 Introduction

These notes are written as supplements to the textbook, “Numerical Linear Algebra” by Lloyd Trefethen and David Bau, for CAAM 454 students. Those who are preparing for the CAAM Ph.D Qualifying exam on Numerical Analysis should do the exercises provided in these notes.

We will mostly follow the notation of the book. For instance, unless otherwise specified, the norm  $\|\cdot\|$  refers to the Euclidian norm.

## 2 Arnoldi – Modified Gram-Schmidt

We consider solving a linear system of equations

$$Ax = b, \tag{1}$$

where we will assume, unless specified otherwise, that  $A$  is  $m$  by  $m$  and complex, i.e.,  $A \in \mathbb{C}^{m \times m}$ , nonsingular, and the right-hand side (RHS) vector

$b \in \mathbb{C}^m$  is nonzero.

The  $n$ -th Krylov subspace generated by  $A$  and  $b$  is

$$\mathcal{K}_n := \langle b, Ab, A^2b, \dots, A^{n-1}b \rangle := \langle p_1, p_2, p_3, \dots, p_n \rangle, \quad (2)$$

where we have introduced the  $p$  vector for the ease of notation.

The “natural” basis,  $\{p_1, p_2, p_3, \dots, p_n\}$ , for  $\mathcal{K}_n$  are generally not orthonormal. The Gram-Schmidt process can be used to generate an orthonormal basis:

$$q_1 = \frac{p_1}{\|p_1\|}, \quad q_{n+1} = \frac{P_{\mathcal{K}_n^\perp} p_{n+1}}{\|P_{\mathcal{K}_n^\perp} p_{n+1}\|}, \quad n = 1, 2, 3, \dots,$$

where  $P_{\mathcal{K}_n^\perp}$  is the projection onto the orthogonal complement space of  $\mathcal{K}_n$ . Specifically,

$$P_{\mathcal{K}_n^\perp} = I - Q_n Q_n^*, \quad \text{and} \quad Q_n = [q_1 \ q_2 \ \dots \ q_n].$$

Invoking the relation  $p_{n+1} = Ap_n$ , we rewrite the Gram-Schmidt process:

$$q_1 = \frac{p_1}{\|p_1\|}, \quad q_{n+1} = \frac{P_{\mathcal{K}_n^\perp} Ap_n}{\|P_{\mathcal{K}_n^\perp} Ap_n\|}, \quad n = 1, 2, 3, \dots, \quad (3)$$

The Arnoldi process can be regarded as a modified Gram-Schmidt process:

$$q_1 = \frac{p_1}{\|p_1\|}, \quad q_{n+1} = \frac{P_{\mathcal{K}_n^\perp} Aq_n}{\|P_{\mathcal{K}_n^\perp} Aq_n\|}, \quad n = 1, 2, 3, \dots, \quad (4)$$

We see that the only difference between the two is the replacement of  $p_n$  by  $q_n$ .

## Exercises

1. Verify the formula in (4).
2. Prove that, until a breakdown occurs, (3) and (4) generate the same sequence of vectors (disregarding possible sign differences).

3. Prove that a breakdown in Arnoldi process happens at the  $n$ -th step of the Arnoldi process if and only if (i)  $Aq_n \in \mathcal{K}_n$ , or (ii)  $\mathcal{K}_p = \mathcal{K}_n$  for all  $p \geq n$ , or (iii)  $\dim(\mathcal{K}_p) = n$  for all  $p \geq n$ .
4. Prove that  $q_{n+1}$  satisfies  $q_{n+1}^* A q_j = 0$  for  $j < n$  (this property is called  $A$ -conjugacy when  $A$  is Hermitian).

### 3 GMRES

The equations

$$AQ_n = Q_{n+1}\tilde{H}_n \quad \text{and} \quad b = \|b\|q_1 \equiv \|b\|Q_{n+1}e_1$$

allow us to write

$$\|r_n\| := \min_{x \in \mathcal{K}_n} \|Ax - b\| = \|b\| \min_{y \in \mathbb{C}^n} \|\tilde{H}_n y - e_1\|, \quad (5)$$

where  $\tilde{H}_n \in \mathbb{C}^{(n+1) \times n}$ , and  $e_1$  is the “first unit vector” in  $\mathbb{R}^{n+1}$ . Upon partitioning, we write the upper Hessenberg matrix  $\tilde{H}_n$  as

$$\tilde{H}_n = \begin{pmatrix} \tilde{H}_{n-1} & h_n \\ 0 \cdots 0 & h_{n+1,n} \end{pmatrix}, \quad (6)$$

where  $h_{n+1,n} > 0$ . Let the  $QR$ -factorization of  $\tilde{H}_n$  be

$$\tilde{H}_n = W_{n+1}R_n \quad (7)$$

where  $W_{n+1} \in \mathbb{C}^{(n+1) \times (n+1)}$  is unitary and  $R_n \in \mathbb{C}^{(n+1) \times n}$  is upper triangular. Clearly, we can write

$$R_n = \begin{pmatrix} \hat{R}_n \\ 0 \cdots 0 \end{pmatrix}, \quad R_n y = \begin{pmatrix} \hat{R}_n y \\ 0 \end{pmatrix},$$

where  $\hat{R}_n$  is  $n$  by  $n$  and nonsingular. Let the first column of  $W_{n+1}^*$  (or the first row of  $\bar{W}_{n+1}$ ) be

$$W_{n+1}^* e_1 = \begin{pmatrix} g \\ \gamma \end{pmatrix}$$

where  $g \in \mathbb{C}^n$  and  $\gamma \in \mathbb{C}$ . Then it is easy to see that since

$$\|\tilde{H}_n y - e_1\|^2 = \|R_n y - W_{n+1}^* e_1\|^2 = \|\hat{R}_n y - g\|^2 + |\gamma|^2,$$

we must have

$$\min_{y \in \mathbb{C}^n} \|\tilde{H}_n y - e_1\| = |\gamma| \equiv |W_{n+1}(1, n+1)|, \quad (8)$$

which is the  $(1, n+1)$ -th element of  $W_{n+1}$  at its north-eastern corner.

### 3.1 Residual Updating

Let us denote the last column of  $W_n$  as

$$w_n = W_n(:, n) \in \mathbb{C}^n, \quad n = 1, 2, 3, \dots$$

From the recursive relation (6), we have

$$W_{n+1} R_n = \begin{bmatrix} W_n R_{n-1} & h_n \\ 0 & h_{n+1, n} \end{bmatrix} = \begin{bmatrix} W_n & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{n-1} & W_n^* h_n \\ 0 & h_{n+1, n} \end{bmatrix}.$$

Hence,

$$R_n = \begin{bmatrix} I_{n-1} & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} R_{n-1}(1:n-1, :) & W_n(:, 1:n-1)^* h_n \\ 0 \cdots 0 & w_n^* h_n \\ 0 \cdots 0 & h_{n+1, n} \end{bmatrix}.$$

where  $G$  is the 2-by-2 Givens rotation matrix

$$G = \begin{bmatrix} \nu_n & \tau_n \\ -\tau_n & \nu_n \end{bmatrix},$$

with elements

$$\tau_n = \frac{h_{n+1,n}}{\sqrt{h_{n+1,n}^2 + |w_n^* h_n|^2}}, \quad \nu_n = \frac{w_n^* h_n}{\sqrt{h_{n+1,n}^2 + |w_n^* h_n|^2}}. \quad (9)$$

On the other hand,

$$W_{n+1} = \begin{bmatrix} W_n & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{n-1} & 0 \\ 0 & G^* \end{bmatrix},$$

which leads to the following recursion:

$$w_{n+1} = \begin{bmatrix} -\tau_n w_n \\ \nu_n^* \end{bmatrix}. \quad (10)$$

Therefore, we have

$$\frac{\|r_n\|}{\|r_{n-1}\|} = \frac{|W_{n+1}(1, n+1)|}{|W_n(1, n)|} = \tau_n, \quad (11)$$

or the residual update formula:

$$\|r_n\| = \tau_n \|r_{n-1}\|. \quad (12)$$

### 3.2 A GMRES Implementation

When  $h_{n+1,n} = 0$ , we know that the solution to  $Ax = b$  is

$$x = Q_n(\|b\|y^*), \quad y^* = \arg \min_{y \in \mathbb{C}^n} \|\tilde{H}y - e_1\|. \quad (13)$$

One simple idea of implementing GMRES would be carrying out Arnoldi iterations until  $h_{n+1,n} = 0$ , then getting solution from (13). This way, one would only solve one least squares problem at the end. However, on a computer,  $h_{n+1,n}$  may never be exactly zero. How about setting a small tolerance for  $h_{n+1,n}$ ? This is difficult too because the relationship between

$h_{n+1,n}$  and the residual is problem-dependent. Any given tolerance may cause termination to occur too early or too late.

Here we present an implementation based on the recursive formula (12) for the residuals. At every iteration, we can update the residual at a cost of  $O(n)$  operations without solving the least square problem. In this implementation, we do not need the matrices  $W_{n+1}$ , except its last column, nor  $R_n$ . Only a single least-squares problem is solved at the end of the algorithm at a cost of  $O(n^2)$  operations (since  $\tilde{H}_n$  is upper Hensenberg), which is dominated by the  $O(mn)$  operations required by forming the solution  $x = Q_n y^*$  in (13) after the least-squares problem is solved.

**Algorithm** (GMRES):

**Initialize:**  $\|r_0\| = \|b\|$ ,  $q_1 = b/\|b\|$ ,  $w_1 = 1$ ,  $\tilde{H}_0 = [ ]$ , and set **tol**  $> 0$ .

**For**  $n = 1, 2, 3, \dots$ ,

1. Do Arnaldi to get  $q_{n+1}$ ,  $h_n$ ,  $h_{n+1,n}$  and update  $\tilde{H}_n$  as in (6).
2. Compute  $\tau_n$  and  $\nu_n$  by the formulas in (9).
3. Update  $w_{n+1}$  by the formula in (10).
4. Update  $\|r_n\|$  by the formula in (12).
5. If  $\|r_n\| > \mathbf{tol} \|b\|$ , go back to step 1.
6. Compute solution  $x$  from (13) and exit.

**End**

The above implementation differs from the standard implementation, given in Saad's book [4], in the order of computations. Both are based on the QR factorization of  $\tilde{H}_n$ . In the standard implementation, the upper

triangular matrix  $R_n$  (the  $R$ -factor of  $\tilde{H}_n$ ) and the right-hand vector  $g$  are explicitly stored and updated, while in our implementation, only a vector  $w$  (the last column of  $W_{n+1}$ ) is stored and updated. On the other hand, at the end of the algorithm, the standard implementation only requires to solve a triangular system of the form  $\hat{R}_n y = g$ , while our implementation required to solve a least-squares problem with the upper Hessenberg matrix  $\tilde{H}_n$ .

The total number of operations required for the two implementations are essentially the same. However, our implementation has a considerably simpler and more compact form, involving no matrix nor matrix operations at every iteration besides those in the Arnoldi process. Hence, it appears easier to understand and to implement by students and other beginners.

### 3.3 GMRES at its Worst

In view of (11) and (9), and noting  $|w_n^* h_n| \leq \|h_n\|$ , we have the estimate

$$\frac{\|r_n\|}{\|r_{n-1}\|} \geq \frac{h_{n+1,n}}{\sqrt{h_{n+1,n}^2 + \|h_n\|^2}}, \quad (14)$$

or equivalently,

$$\frac{\|r_n\|}{\|r_{n-1}\|} \geq \left\| P_{\mathcal{K}_n^\perp} \frac{Aq_n}{\|Aq_n\|} \right\|. \quad (15)$$

The left-hand sides are the best possible reduction factor in the residual after each and every iteration. Obviously, when  $Aq_n$  is almost orthogonal to  $\mathcal{K}_n$ , or when  $\|h_n\|$  is much less than  $h_{n+1,n}$ , little reduction can be achieved.

The inequality (14) reveals that the best possible convergence rate of GMRES is completely determined by the Hessenberg factor  $H$  of the matrix  $A$ , i.e.,  $A = QHQ^*$ , where the first column  $q_1$  in  $Q$  is a normalized initial residual vector. Let us assume that we solve  $Ax = b$  starting from  $x_0 = 0$ , hence  $q_1 = \pm b/\|b\|$ .

It is easy to observe from (14) that the absolutely worst matrix possible for GMRES is  $A = QHQ^*$  where the upper Hessenberg matrix  $H$  has the form

$$H = \begin{bmatrix} 0 & & & \times \\ \times & 0 & & \times \\ & \times & \ddots & \vdots \\ & & \ddots & 0 & \times \\ & & & \times & \times \end{bmatrix}_{m \times m}, \quad (16)$$

where, for non-singularity, the  $\times$ 's on the sub-diagonal and at the  $(1, m)$  position should be nonzeros.

For this  $H$  matrix, we have

$$h_{n+1,n} \neq 0, \quad \|h_n\| = 0, \quad n < m.$$

Therefore,  $\tau_n \equiv 1$  for  $n < m$ , and

$$\|r_n\| = \|b\|, \quad n < m.$$

That is, GMRES will make no progress at all until the very last iteration where  $\|r_m\| = 0$ . The iterates are identically zero:

$$x_0 = x_1 = \cdots = x_{m-1} = 0,$$

except for  $x_m = x^*$ , the solution. This is when GMRES at its worst!

We should point out that this worst behavior of GMRES does not seem a “measure-zero” phenomenon. In our experience, GMRES behaves almost as badly for matrices  $A = Q(H + E)Q^*$  where  $H$  is of the form in (16) and  $E$  is a sufficiently small perturbation (even starting from random initial guesses).

## Exercises

1. Prove (8).
2. Prove (10).
3. Prove (14) and (15).

## 4 Restarted GMRES( $t$ )

GMRES becomes more and more expensive as the iteration number increases. A popular fix to this problem is to restart GMRES after every  $t$  iterations with the current residual as the right-hand side. The residual norm will be monotonically non-increasing.

GMRES( $t$ ) can work really well in most cases, sometimes even better than GMRES in terms of the total number of (inner) iterations [1]. However, stalling can happen from time to time when the residual norm stops decreasing at a meaningful rate or at all. What is the best strategy, if there is one, to choose the restarting frequency  $t$ ? The following quote is from our own CAAM professor Embree, an expert [2] in this field:

Optimal selection of the GMRES restart parameter is one of the most perplexing open questions in the field.

What we can say for sure is that for whatever choice of restarting parameter  $t < m$ , non-convergence can happen, as will be shown next.

### 4.1 All Restarts Fail in the Worst Case

Let us again consider the worst case for GMRES where  $A = QHQ^*$ , and  $H$  is of the form (16). Observe that, starting from  $q_1 = r_0 = b/\|b\|$ , GMRES( $t$ ),

$t < m$ , restarts from  $r_{kt}$  after every  $p$  inner iterations. It is easy to verify that

$$\arg \min_{x \in \mathcal{K}_p} \|Ax - b\| = 0 \Rightarrow x_{kt} = 0, r_{kt} = b, \forall p < m.$$

Therefore, in all outer iterations GMRES( $t$ ) always restarts from the same point with  $x_{kt} = 0$  and  $r_{kt} = b$ . Obviously, all the iterates, whether inner or outer, are identically zero for ever.

## Exercises

1. Derive an explicit formula for GMRES(1) in the form of  $x_{n+1} = x_n + d_n$ . What is the main difference, if any, between GMRES(1) and speediest descent method for  $\min \|Ax - b\|^2$ ?

## References

- [1] Mark Embree. The Tortoise and the Hare Restart GMRES. *SIAM Rev.* 45 (2003) 259-266.
- [2] Mark Embree. Private communications. 2007.
- [3] Anne Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, 1997.
- [4] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* 2nd Ed., SIAM, 2003.