CAAM 454/554: Afternotes on Krylov-Subspace Iterative Methods

Yin Zhang (draft)
CAAM, Rice University, Houston, TX 77005

February 3, 2007
Revised 02/24/2011, 01/29/2013

1 Introduction

These notes are written as supplements to the textbook, "Numerical Linear Algebra" by Lloyd Trefethen and David Bau, for CAAM 454/554 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

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

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

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

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

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

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

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

$$q_1 = \frac{p_1}{\|p_1\|}, \quad q_{n+1} = \frac{P_{\mathcal{K}_n^\perp} A p_n}{\|P_{\mathcal{K}_n^\perp} A p_n\|}, \quad n = 1, 2, 3, \cdots, \tag{3}$$

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

$$q_1 = \frac{p_1}{\|p_1\|}, \quad q_{n+1} = \frac{P_{\mathcal{K}_n^\perp} A q_n}{\|P_{\mathcal{K}_n^\perp} A q_n\|}, \quad n = 1, 2, 3, \cdots, \tag{4}$$

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

2.1 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-conjugatcy when A is Hermitian).

3 GMRES

GMRES Method can be summarized into a one-liner:

$$x_n = \arg\min_{x \in \mathcal{K}_n} \|b - Ax\|_2.$$

Since $x = Q_n y$ for $x \in \mathcal{K}_n$, 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

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

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

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

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

$$\tilde{H}_n = W_{n+1} R_n \tag{7}$$

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

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

which is the modulus of 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

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

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

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

with elements

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

where $\tau_n \in [0,1]$ and ν_n may be complex.

On the other hand, letting $\hat{W}_n$ be the sub-matrix consisting of the first $n-1$ columns of W_n and w_n be the last column, then

$$W_{n+1} = \begin{bmatrix} W_n & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{n-1} & 0 \\ 0 & G^* \end{bmatrix} = \left(\begin{bmatrix} \hat{W}_n \\ 0 \end{bmatrix} \quad \begin{bmatrix} w_n & 0 \\ 0 & 1 \end{bmatrix} G^* \right),$$

which leads to the following recursion:

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

Therefore, we have

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

or the residual update formula:

$$\|r_n\| = \tau_n \, \|r_{n-1}\|. \tag{12}$$

3.2 A GMRES Implementation

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

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

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 $n = 1, 2, 3, \cdots,$

1. Do Arnaldi to get q_{n+1}, h_n, $h_{n+1,n}$ and update $\tilde{H}_n$ as in (6).

2. Compute τ_n and ν_n by the formulas in (9).

3. Update to w_{n+1} by the formula in (10).

4. Update $\|r_n\|$ by the formula in (12).

5. If $\|r_n\| > \texttt{tol}\,\|b\|$, 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-squrares problem with the upper Hensenberg 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 $|w_n^* h_n| \le \|h_n\|$, we have the estimate

$$\frac{\|r_n\|}{\|r_{n-1}\|} \ge \frac{h_{n+1,n}}{\sqrt{h_{n+1,n}^2 + \|h_n\|^2}}, \tag{14}$$

or equivalently,

$$\frac{\|r_n\|}{\|r_{n-1}\|} \ge \left\| P_{\mathcal{K}_n^\perp} \frac{Aq_n}{\|Aq_n\|} \right\|. \tag{15}$$

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 then $h_{n+1,n}$, 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

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

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

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

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).

3.4 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, somethimes 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

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

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.

Although the worst-case picture appears quite gloomy, the occurrence of the worst-cases is really an unlikely event that one hardly encounters in practical applications. In practice, GMRES(t) generally works pretty well.

4.2 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 $\min \|Ax - b\|^2$?

5 Conjugate Gradient Method

Now we consider linear systems $Ax = b$ with $A \succ 0$, meaning that the matrix $A \in \mathbb{R}^{m \times m}$ is symmetric and positive definite (i.e., $x^T Ax > 0$ for all nonzero $x \in \mathbb{R}^m$).

5.1 Framework

We will continue to follow the GMRES idea of minimizing the residual $r = b - Ax$ in the Krylov subspaces $\mathcal{K}_n(A, b)$ with increasing n. However, we will change the norm used to measure residual sizes.

Definition 1. *For any $m \times m$ symmetric positive definite matrix $W \succ 0$, we define a weighted 2-norm for $\mathbb{R}^m$:*

$$\|x\|_W \triangleq \sqrt{x^T W x}, \quad \forall\, x \in \mathbb{R}^m.$$

Conjugate Gradient (CG) Method can be summarized into a one-liner:

$$x_n = \arg\min_{x \in \mathcal{K}_n} \|b - Ax\|_{A^{-1}}. \tag{17}$$

It is easy to verify that for $x^* = A^{-1}b$,

$$\|b - Ax\|_{A^{-1}}^2 = \|x - x^*\|_A^2 = 2\phi(x) + \text{const} \tag{18}$$

where $\phi(x)$ is the quadratic function

$$\phi(x) \triangleq \frac{1}{2} x^T A x - b^T x. \tag{19}$$

Therefore, CG is nothing but

$$x_n = \arg\min_{x \in \mathcal{K}_n} \phi(x). \tag{20}$$

The gradient and the Hessian of $\phi(x)$ are, respectively,

$$\nabla\phi(x) = Ax - b \quad \text{and} \quad \nabla^2\phi(x) = A.$$

5.2 A-Conjugacy

Instead of using orthonormal bases for $\mathcal{K}_n$, we will use so-called A-conjugate bases. The benefit of using such bases will become clear soon.

Definition 2. *A set of nonzero vectors $\{p_0, p_1, \cdots, p_{n-1}\} \subset \mathbb{R}^m$ is said to be A-conjugate with respect to a symmetric positive definite matrix $A \in \mathbb{R}^{m \times m}$ if the vectors satisfy*

$$p_i^T A p_j = 0, \quad \forall\, i \neq j.$$

A-conjugacy allows the following recursion:

Lemma 1. *Let $\{p_0, p_1, \cdots, p_{n-1}\}$ be an A-conjugate basis of $\mathcal{K}_n$ for $n \geq 1$, and $\{x_n\}$ be defined as in (20). In addition, let $x_0 = 0$. Then for $n \geq 1$*

$$x_n = x_{n-1} + \alpha_n p_{n-1}, \tag{21}$$

where

$$\alpha_n = \frac{r_{n-1}^T p_{n-1}}{p_{n-1}^T A p_{n-1}} = \arg\min_{\alpha \in \mathbb{R}} \phi(x_{n-1} + \alpha p_{n-1}). \tag{22}$$

In this case, minimizing $\phi(x)$ in $\mathcal{K}_n$ is equivalent to minimizing $\phi(x)$ in the direction of p_{n-1} starting from x_{n-1}.

5.3 Exercises

1. Let A be symmetric. Prove, only using the definition

$$A \succ 0 \iff x^T Ax > 0,\ \forall x \neq 0,$$

that (a) if $A \succ 0$, then A is nonsingular and $A^{-1} \succ 0$; (b) the two equalities in (18).

2. Prove that A-conjugacy implies linear independence.

3. Prove Lemma 1.

5.4 Descent Methods

Definition 3. *Let $\psi : \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. A vector $p \in \mathbb{R}^m$ is a descent direction of ψ at x if*

$$\nabla\psi(x)^T p < 0.$$

The steepest descent direction of ψ at x is $p = -\nabla\psi(x)$ if it is nonzero.

Given x and p, if we define

$$f(\alpha) = \psi(x + \alpha p),$$

then it is clear that f is decreasing at $\alpha = 0$ if $f'(0) = \nabla\psi(x)^T p < 0$.

The descent method framework (with exact line search) is

$$\begin{cases} \alpha_n &= \arg\min_{\alpha\in\mathbb{R}} \psi(x_{n-1} + \alpha p_{n-1}), \\ x_n &= x_{n-1} + \alpha_n p_{n-1}. \end{cases} \tag{23}$$

Naturally, the *steepest descent method* corresponds to $p = -\nabla\psi(x)$.

Since α_n is a minimum of $f(\alpha)$, there must hold $f'(\alpha_n) = 0$; i.e.,

$$\nabla\psi(x_n)^T p_{n-1} = 0. \tag{24}$$

For the quadratic function $\phi(x)$ defined in (19), the steepest descent direction is the residual at each iteration since

$$p = -\nabla\phi(x) = b - Ax = r.$$

When applied to such a quadratic function $\phi(x)$, the steepest descent method takes the following

form.

Steepest Descent Method

Set $x_0 = 0$ and $p_0 = r_0 = b$.

for $k = 1, 2, \cdots$, **do**

$\alpha_n = r_{n-1}^T p_{n-1} / p_{n-1}^T A p_{n-1}$;

$x_n = x_{n-1} + \alpha_n p_{n-1}$;

$r_n = r_{n-1} - \alpha_n A p_{n-1}$;

$p_n = r_n$.

end

It can be directly derive from the algorithm steps that

$$\frac{\phi(x_n) - \phi(x_*)}{\phi(x_{n-1}) - \phi(x_*)} = 1 - \frac{1}{\gamma(r_{n-1})}, \tag{25}$$

where

$$\gamma(d) = \frac{(d^T A d)(d^T A^{-1} d)}{(d^T d)^2}. \tag{26}$$

By Kantorovich Inequality,

$$\gamma(d) \leq \frac{(\lambda_{\max}(A) + \lambda_{\min}(A))^2}{4\lambda_{\max}(A)\lambda_{\min}(A)}, \quad \forall d \neq 0, \tag{27}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the largest and the smallest eigenvalues of A, respectively. Hence, we can derive the convergence rate for the steepest descent method to be

$$\frac{\phi(x_n) - \phi(x_*)}{\phi(x_{n-1}) - \phi(x_*)} \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2, \tag{28}$$

where $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the condition number of A in Euclidean norm.

5.5 Conjugate Gradient Method

CG is a descent method as the steepest descent method is. It differs from the latter only in the choice of search directions. In the last step for p_n, CG adds a portion of the previous search direction βp_{n-1} to r_n to form the next search direction $p_n = r_n + \beta p_{n-1}$. The parameter β is chosen to make p_n and p_{n-1} A-conjugate. That is, $(r_n + \beta p_{n-1})^T A p_{n-1} = 0$, giving

$$\beta_n = \frac{-r_n^T A p_{n-1}}{p_{n-1}^T A p_{n-1}}. \tag{29}$$

From Lemma 1, we know that A-conjugacy reduces Krylov subspace minimization of $\phi(x)$ to a simple one-dimensional minimization in the direction of p_{n-1} emitting from x_{n-1}, producing exactly

the same update as in the steepest descent method. A preliminary form of CG is as follows.

CG Method

Set $x_0 = 0$ and $p_0 = r_0 = b$.

for $k = 1, 2, \cdots$, **do**

$\alpha_n = r_{n-1}^T p_{n-1} / p_{n-1}^T A p_{n-1}$;

$x_n = x_{n-1} + \alpha_n p_{n-1}$;

$r_n = r_{n-1} - \alpha_n A p_{n-1}$;

$\beta_n = -r_n^T A p_{n-1} / p_{n-1}^T A p_{n-1}$;

$p_n = r_n + \beta_n p_{n-1}$.

end

Obviously, when $\beta_n = 0$, CG reduces to the steepest descent method.

Equivalent but more refined expressions for the two parameters α_n and β_n can be derived from the properties of the CG method (TB Theorem 38.1); i.e.,

$$\alpha_n = \frac{r_{n-1}^T r_{n-1}}{p_{n-1}^T A p_{n-1}}, \quad \beta_n = \frac{r_n^T r_n}{r_{n-1}^T r_{n-1}}, \tag{30}$$

which are slightly more efficient to compute.

5.6 Exercises

1. Prove (24), and then use it to show that the search direction p_n in CG is a descent direction of the quadratic $\phi(x)$ at x_n.
2. Prove the equality (25) for the steepest descent method.
3. Given (25)-(27), prove the convergence rate estimate (28) for the steepest descent method.
4. Use Theorem 38.1 in TB textbook to derive (30) from their expressions in the preliminary form of the CG method.

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.