Motivations and realizations of Krylov subspace methods for large sparse linear systems

Zhong-Zhi Bai

PII: S0377-0427(15)00037-0
DOI: http://dx.doi.org/10.1016/j.cam.2015.01.025
Reference: CAM 9979

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 14 April 2014
Revised date: 7 October 2014

Please cite this article as: Z.-Z. Bai, Motivations and realizations of Krylov subspace methods for large sparse linear systems, *Journal of Computational and Applied Mathematics* (2015), http://dx.doi.org/10.1016/j.cam.2015.01.025

Motivations and Realizations of Krylov Subspace Methods for Large Sparse Linear Systems

Zhong-Zhi Bai

Qian Weichang College, Shanghai University
Shanghai 200436, P.R. China

and

State Key Laboratory of Scientific/Engineering Computing
Institute of Computational Mathematics and Scientific/Engineering Computing
Academy of Mathematics and Systems Science
Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, P.R. China
Email: bzz@lsec.cc.ac.cn

October 8, 2014

Abstract

We briefly introduce typical and important direct and iterative methods for solving systems of linear equations, concretely describe their fundamental characteristics in viewpoints of both theory and applications, and clearly clarify the substantial differences among these methods. In particular, the motivations of searching the solution of a linear system in a Krylov subspace are described and the algorithmic realizations of the *generalized minimal residual* (**GMRES**) method are shown, and several classes of state-of-the-art algebraic preconditioners are briefly reviewed. All this is useful for correctly, deeply and completely understand the application scopes, theoretical properties and numerical behaviors of these methods, and is also helpful in designing new methods for solving systems of linear equations.

Keywords: linear system, direct method, iterative method, Krylov subspace, preconditioning.

AMS(MOS) Subject Classifications: 65F10, 65F15, 65C40; CR: G1.3.

1 Introduction

The system of linear equations

$$Ax = b, \quad \text{with} \quad A \in \mathbb{R}^{n\times n} \quad \text{nonsingular and} \quad x, b \in \mathbb{R}^n, \tag{1}$$

can be solved efficiently by either a direct or an iterative method [40, 43, 21, 1, 25]. Roughly speaking, the direct methods are based on the lower-upper triangular and the orthogonal-triangular factorizations, or in brief, the LU (or the Gaussian elimination) and the QR factorizations [25], and the iterative methods are based on the matrix splittings [40, 43, 25, 11, 9] and the Krylov subspaces [1, 25, 5, 33]. These two classes of methods are principally different but computationally dependent. More specifically, an iterative method can be used to refine an approximate solution computed by a direct method, and a direct method can be employed to precondition an iterative method. In particular, in the Krylov subspace iteration methods the orthogonal basis of the subspace is often computed through a stable variant of the QR factorization [39]. Therefore, a technical and skillful combination of direct and iterative methods can produce fast, stable and accurate linear solvers for the linear system (1).

While numerical stability and computational complexity are main issues in theoretically analyzing the direct methods, and balanced scaling, proper pivoting and effective ordering are essential strategies in their practical implementations [21, 23], a challenge in theoretical study of the Krylov subspace iteration methods is convergence analysis of the iteration sequences, and a major difficulty in practical usage of these methods is algorithmic construction of high-quality preconditioner and algebraic analysis of the corresponding preconditioned matrix [5, 33, 35].

In this paper, we will briefly review several typical and important direct and iterative methods that are economical and effective for solving the linear system (1). The motivations of searching the solution x_* of the linear system (1) in a Krylov subspace are described in detail, and the algorithmic realizations of GMRES [34] are shown deliberately. Also, we discuss convergence properties of the Krylov subspace iteration methods such as GMRES when the matrix A is symmetric and when it is nonsymmetric but diagonalizable, and review several classes of effective preconditioners such as those based on *incomplete LU* (**ILU**), *incomplete QR* (**IQR**), sparse approximate inverses, matrix splitting iterations, and algebraic multilevel and multigrid iteration techniques. All this could be useful for correctly, deeply and completely understand the application scopes, theoretical properties and numerical behaviors of these methods, and should be helpful in designing new methods for solving systems of linear equations.

2 The Direct Methods

In the Gaussian elimination method, we successively operate the Gauss transforms one a time on the expanded matrix $[A \mid b]$, and finally obtain the target matrix $[I \mid x_*]$, that is, $[A \mid b] \to [I \mid x_*]$ in symbolic, where I is the identity matrix. This method can solve only one system a time, requiring approximately the storage $n^2 + n$ (n^2 for the coefficient matrix A and n for the right-hand side b) and the operations $\frac{2}{3}n^3$. The methodology of the LU factorization is a little bit different from the Gaussian elimination, which first factorizes the coefficient matrix A into the product of a lower-triangular matrix L and an upper-triangular matrix U, i.e., $A = LU$, and then computes the exact solution x_* of the linear system (1) through a forward elimination and a backward substitution. The LU factorization method can solve many linear systems having the same coefficient matrix A but different right-hand sides b a time.

For the QR factorization method, we first factorize the coefficient matrix A into a product of an orthogonal matrix Q and an upper-triangular matrix R, obtaining $A = QR$, and then compute the exact solution x_* of the linear system (1) through a backward substitution due to

$Rx = Q^T b$. The QR factorization requires approximately the storage $n^2 + n$ and the operations $2n^3$. Here and in the sequel, we use $(\cdot)^T$ to indicate the transpose of either a vector or a matrix.

Besides the differences in storage and operation mentioned above, it has been proved that the LU factorization exists only for strictly diagonal dominant matrices and symmetric positive definite matrices, but the QR factorization may exist for any matrix even for a rectangular one. Hence, the QR factorization can be employed to solve the linear least-squares problems via, e.g., the seminormal equation $R^T Rx = A^T b$; see [36]. Moreover, if $A \in \mathbb{R}^{n\times n}$ is sparse, then both L and U may be also sparse, but Q and R could be dense. Hence, each of these two factorizations has its pros and cons.

As we have known, Givens rotation, Householder reflection and Gram-Schmidt orthogonalization are three classical and typical tools for computing a QR factorization for a given matrix. Below we review the classical Gram-Schmidt orthogonalization process and its stabilized modification, in which the latter is the elementary ingredient of the Krylov subspace iteration methods.

Let

$$A = [a_1, a_2, \ldots, a_n], \quad Q = [q_1, q_2, \ldots, q_n]$$

and

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix},$$

where a_i and q_i are the i-th columns of the matrices A and Q, respectively. Then $A = QR$ or

$$[a_1, a_2, \ldots, a_n] = [q_1, q_2, \ldots, q_n] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}$$

is equivalent to

$$\begin{cases} a_1 = r_{11}\, q_1, \\ a_2 = r_{12}\, q_1 + r_{22}\, q_2, \\ a_3 = r_{13}\, q_1 + r_{23}\, q_2 + r_{33}\, q_3, \\ \cdots \\ a_n = r_{1n}\, q_1 + r_{2n}\, q_2 + \cdots + r_{nn}\, q_n, \end{cases}$$

which straightforwardly results in the following orthogonalization process, called the classical Gram-Schmidt process.

The Classical Gram-Schmidt Process

For $j = 1 : n$

$v_j = a_j$

For $i = 1 : j - 1$

$r_{ij} = q_i^T a_j$

$v_j = v_j - r_{ij}\, q_i$

$r_{jj} = \|v_j\|$

$q_j = v_j / r_{jj}$

The classical Gram-Schmidt process is numerically unstable. A stabilized modification, called the modified Gram-Schmidt process, is described in the following.

The Modified Gram-Schmidt Process

$v_1 = a_1 / \|a_1\|$

For $j = 1 : n$

$\widetilde{v}_j = a_j$

For $i = 1 : j - 1$

$\widetilde{v}_j = \widetilde{v}_j - (v_i^T \widetilde{v}_j)\, v_i$

$v_j = \widetilde{v}_j / \|\widetilde{v}_j\|$

We remark that the modified Gram-Schmidt process is, philosophically speaking, an application of the idea of the Gauss-Seidel sweep used for iteratively solving linear systems.

Of course, besides the LU and the QR factorization methods stated above, the famous Gram rule gives the most beautiful analytic formula for the solution x_* of the linear system (1). Precisely speaking, in terms of the determinants of the matrices A and

$$A_j = [a_1, \ldots, a_{j-1}, b, a_{j+1}, \ldots, a_n], \quad j = 1, 2, \ldots, n,$$

the j-th element $x_*^{[j]}$ of x_* is given by

$$x_*^{[j]} = \frac{\det(A_j)}{\det(A)}, \quad j = 1, 2, \ldots, n, \tag{2}$$

where $\det(\cdot)$ denotes the determinant of the corresponding matrix. As is well known, the cost of this formula is tremendous like $\mathcal{O}(n^2\, n!)$, so it is practically prohibitive especially when the matrix A is large and sparse. However, by making use of the LU or the QR factorization we propose here a practical implementation for the Gram rule. To this end, we only consider the general case that A is nonsingular and nonsymmetric, as the special case that A is symmetric positive definite can be treated analogously by utilizing the Cholesky factorization [25] of the matrix A instead of LU or QR. Let $A = LU$ be the LU factorization, $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)^T$

be the j-th unit basis vector in $\mathbb{R}^n$, and $v_j = e_j - A^{-1}b$. Then we have $Ae_j = a_j$ and

$$A_j = A - (a_j - b)e_j^T = A(I - v_j e_j^T).$$

So

$$\det(A_j) = \det(A) \cdot \det(I - v_j e_j^T).$$

Because

$$(I - v_j e_j^T)v_j = (1 - e_j^T v_j)v_j = (e_j^T A^{-1}b)v_j$$

and

$$(I - v_j e_j^T)e_i = e_i, \quad \text{for} \quad i \neq j,$$

the eigenvalues of the matrix $I - v_j e_j^T$ are 1 with multiplicity $n-1$ and $e_j^T A^{-1}b$, which implies that

$$\det(I - v_j e_j^T) = e_j^T A^{-1}b.$$

It follows immediately from (2) that

$$x_*^{[j]} = \det(I - v_j e_j^T) = e_j^T A^{-1}b.$$

As a result, we obtain the following procedure for computing x_*.

A Practical Implementation of the Gram Rule

`Compute` $A = LU$

`Solve` $Lv = b$ `and` $Ux_* = v$

This procedure is the same as the LU factorization method. It implements the Gram rule in $\frac{2}{3}n^3$ operations, in the same cost as that of either the LU or the Gaussian elimination. Alternatively, using the QR instead of the LU factorization of the matrix A we can analogously obtain a corresponding practical implementation of the Gram rule, too.

3 The Iterative Methods

The successive relaxation methods and the Krylov subspace methods are two basic classes of iteration methods aiming to solve large sparse linear systems of the form (1). The former is often parameter-dependent and definitely breaks down when one diagonal entry of the matrix A is zero; its construction is perceptual in the sense that one entry of the residual vector is annihilated at each step by solving one variable from one equation. However, the latter is often parameter-free and possibly breaks down due to various reasons; its construction is rational in the sense that the residual vector is minimized at each step (e.g., in the Euclidean or the energy norm). Representatives of the successive relaxation iteration methods are Jacobi,

Gauss-Seidel, SOR (successive overrelaxation) and their symmetric variants [28, 11, 27, 19], and examples of the Krylov subspace iteration methods are CG (conjugate gradient), MINRES (minimal residual), Bi-CGSTAB (stabilized bi-conjugate gradient) and GMRES (generalized minimal residual) [34, 1, 25]. For more references, we refer to [9, 18].

The elementary motivations of these two classes of iteration methods are the same, that is, from the given coefficient matrix $A \in \mathbb{R}^{n\times n}$ and right-hand side $b \in \mathbb{R}^n$, how can we construct a sequence $\{x_k\}$ such that it approximates the solution x_* of the linear system $Ax = b$ rapidly, accurately and stably?

For the Krylov subspace iteration methods, an important feature is that at each iteration only one matrix-vector multiplication and a small number of vector operations (dot products and vector updates) are required. For sparse or structured matrices, the matrix times vector product may be efficiently computed and so the main issue concerning the overall computational work in the iterative solution of a linear system with such methods is the number of iterations it takes for convergence to an acceptable accuracy. For the successive relaxation methods, besides one matrix-vector multiplication they also require to solve one or two triangular linear sub-systems at each iteration step.

A basic and practical strategy of the Krylov subspace iteration methods is as follows: Find

$$x_1 \in \mathrm{span}\{b\}, \quad x_2 \in \mathrm{span}\{b, Ab\}, \ \ldots, \ x_k \in \mathrm{span}\{b, Ab, \ldots, A^{k-1}b\}, \ \ldots$$

by certain prescribed rule so that $\lim\limits_{k\to+\infty} x_k = x_*$. Here the linear subspace

$$\mathcal{K}_k(A, b) = \mathrm{span}\{b, Ab, \ldots, A^{k-1}b\}$$

is called the k-th order Krylov subspace. Then two questions naturally arise:

(i) why do we use a Krylov subspace to construct an iterative method?

(ii) how good an approximate solution is contained in a Krylov subspace?

For Question (i), a traditional reasoning is given via the Richardson extrapolation iteration

$$x_{k+1} = x_k + \omega(b - Ax_k), \qquad k = 0, 1, 2, \ldots,$$

where ω is an iteration parameter. As

$$x_k = (I - \omega A)x_{k-1} + \omega b = \sum_{j=0}^{k-1} (I - \omega A)^j \omega b \in \mathcal{K}_k(A, b)$$

for $x_0 = 0$, it holds that

$$x_* \in \mathcal{K}_\infty(A, b)$$

provided the iteration sequence $\{x_k\}$ is convergent. Alternatively, another reasoning can be given through the minimum-degree polynomial of A; see, e.g., [30]. Let $\phi_k(A)$ be the minimum-degree polynomial of the matrix A with the degree k, and denote

$$\phi_k(A) = \alpha_0 I + \alpha_1 A + \cdots + \alpha_k A^k.$$

Then we know $\alpha_0 \neq 0$ from the nonsingularity of the matrix A and $\alpha_k \neq 0$ from the minimum-degree property of the polynomial $\phi_k(A)$. It follows straightforwardly from $\phi_k(A) = 0$ that

$$A(\alpha_1 I + \alpha_2 A + \cdots + \alpha_k A^{k-1}) = -\alpha_0 I,$$

or equivalently,

$$A^{-1} = -\frac{1}{\alpha_0}(\alpha_1 I + \alpha_2 A + \cdots + \alpha_k A^{k-1}). \tag{3}$$

Therefore,

$$\begin{aligned} x_* &= A^{-1}b = -\frac{1}{\alpha_0}(\alpha_1 I + \alpha_2 A + \cdots + \alpha_k A^{k-1})b \\ &\in \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\} = \mathcal{K}_k(A, b). \end{aligned}$$

Note that these two explanations show distinguishable facts: The former implies that for any fixed k the iterate $x_k \in \mathcal{K}_k(A, b)$ so that $x_* \in \mathcal{K}_\infty(A, b)$, but the latter exhibits that $x_* \in \mathcal{K}_k(A, b)$ for the k being the degree of the minimum polynomial of the matrix A.

An answer to Question (ii) is, however, more involved. From $x_k \in \mathcal{K}_k(A, b)$ we have

$$r_k := b - Ax_k \in b - A\mathcal{K}_k(A, b) \subseteq \mathcal{K}_{k+1}(A, b),$$

or in other words,

$$r_k = \sum_{j=0}^{k} \alpha_j A^j b, \quad \text{with} \quad \alpha_0 = 1.$$

Define a polynomial

$$\Psi_k(t) = \sum_{j=0}^{k} \alpha_j t^j, \quad \text{with} \quad \Psi_k(0) = 1.$$

Then we have

$$r_k = \Psi_k(A)\, b.$$

When the matrix A is symmetric, by writing the eigendecomposition of A as $A = Q\Lambda Q^T$, with $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ being a diagonal matrix of the eigenvalues and Q being an orthogonal matrix of the eigenvectors of the matrix A, we can further obtain

$$\|r_k\| \leq \|Q\, \Psi_k(\Lambda)\, Q^T b\| \leq \|\Psi_k(\Lambda)\| \cdot \|b\| \leq \min_{\Psi_k(0)=1} \max_{1 \leq i \leq n} |\Psi_k(\lambda_i)| \cdot \|b\|.$$

Here and in the sequel we use $\|\cdot\|$ to denote the Euclidean norm of either a vector or a matrix. The important implication is that the Euclidean norm of the residual (for the worst-case right-hand side) is completely determined by the eigenvalues of the matrix A, and we have at least intuitive ideas of what constitute good and bad eigenvalue distributions. More precisely, for a real symmetric matrix, convergence depends only on its eigenvalues: if there are only a few distinct

eigenvalues or they are sufficiently clustered away from the origin then there are polynomials of low degree which will be small at the eigenvalues. At each additional iteration the degree increases by one and so reasonable accuracy is quickly achieved in such cases [1, 16].

And when A is nonsymmetric but diagonalizable, by writing the eigendecomposition of A as $A = V\Lambda V^{-1}$, with Λ being a diagonal matrix of the eigenvalues and V being a nonsingular matrix of the eigenvectors of the matrix A, we can obtain

$$\|r_k\| \leq \|V\,\Psi_k(\Lambda)\,V^{-1}b\| \leq \kappa(V) \min_{\Psi_k(0)=1} \max_{1\leq i\leq n} |\Psi_k(\lambda_i)| \cdot \|b\|.$$

In this case, the Euclidean norm of the residual is determined not only by the eigenvalues, but also by the eigenvectors of the matrix A. Of course, if there are only a few distinct eigenvalues or they are sufficiently clustered away from the origin then there are polynomials of low degree which will be small at the eigenvalues. Provided that the linearly independent eigenvectors are complete and the matrix formed by these eigenvectors is well-conditioned, at each additional iteration the degree increases by one and so reasonable accuracy is quickly achieved in such cases. Then an interesting open question is to determine when an ill-conditioned eigenvector matrix implies poor convergence for a Krylov subspace method and when it simply means that the above bound is a large overestimate.

In general, when A is nonsymmetric and non-diagonalizable, we refer the readers to [5] for a different treatment, and to [5, 38] and the references therein for more estimates on the Euclidean norm of the residual.

In terms of both theory and practice, we then have to ask how a good approximation from the Krylov subspace can be computed with a moderate amount of work and storage? A standard approach answering this question is described in the following.

Let $x_0 \in \mathbb{R}^n$ be an initial vector. Then the linear system (1) is equivalent to $A(x_* - x_0) = r_0$. It follows from (3) that

$$x_* - x_0 = A^{-1}r_0 \in \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\} = \mathcal{K}_k(A, r_0)$$

or

$$x_* \in x_0 + \mathcal{K}_k(A, r_0) \quad \text{and} \quad r = b - Ax_* \in r_0 - A\,\mathcal{K}_k(A, r_0).$$

Let $x_k \in \mathbb{R}^n$ be the k-th approximation to x_* such that

$$x_k \in x_0 + \mathcal{K}_k(A, r_0).$$

Then we have

$$r_k \in r_0 - A\,\mathcal{K}_k(A, r_0).$$

Obviously, we need first to compute an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, r_0)$ by using, for example, the modified Gram-Schmidt process or the so-called Arnoldi process.

The Arnoldi Process

Given q_1 satisfying $\|q_1\| = 1$

For $j = 1, 2, \ldots, k-1$

$\tilde{q}_{j+1} = Aq_j$

For $i = 1 : j$

$h_{ij} = q_i^T \tilde{q}_{j+1}$

$\tilde{q}_{j+1} = \tilde{q}_{j+1} - h_{ij} q_i$

$h_{j+1,j} = \|\tilde{q}_{j+1}\|$

$q_{j+1} = \tilde{q}_{j+1} / h_{j+1,j}$

Denote by

$$Q_k = [q_1, q_2, \ldots, q_k] \in \mathbb{R}^{n \times k}.$$

Then the Arnoldi process yields

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} \xi_k^T = Q_{k+1} H_{k+1,k},$$

where

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ & h_{32} & \ddots & \cdots & h_{3k} \\ & & \ddots & \ddots & \vdots \\ & & & h_{k,k-1} & h_{kk} \end{bmatrix}$$

and

$$H_{k+1,k} = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ & h_{32} & \ddots & \cdots & h_{3k} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & h_{kk} \\ & & & & h_{k+1,k} \end{bmatrix},$$

with $\xi_k = (0, 0, \ldots, 0, 1)^T \in \mathbb{R}^k$ being the last unit basis vector in $\mathbb{R}^k$. As a result, it holds that

$$Q_k^T A Q_k = H_k.$$

As for the GMRES method, computing an approximate x_k such that $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ is equivalent to computing a vector $y_k \in \mathbb{R}^k$ such that

$$x_k = x_0 + Q_k y_k \quad \text{or} \quad r_k = r_0 - AQ_k y_k.$$

Let $\beta = \|r_0\|$ and $\xi = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$. Then from

$$Q_{k+1}\xi = q_1 = \frac{r_0}{\|r_0\|} = \frac{r_0}{\beta}$$

we have

$$\begin{aligned}
\min \|r_k\| &= \min_{y \in \mathbb{R}^k} \|r_0 - AQ_k y\| \\
&= \min_{y \in \mathbb{R}^k} \|r_0 - Q_{k+1} H_{k+1,k} y\| \\
&= \min_{y \in \mathbb{R}^k} \|Q_{k+1}(\beta\xi - H_{k+1,k} y)\| \\
&= \min_{y \in \mathbb{R}^k} \|\beta\xi - H_{k+1,k} y\|.
\end{aligned}$$

This property straightforwardly leads to the GMRES method described below.

The GMRES Method

- Given x_0, compute $r_0 = b - Ax_0$ and set $q_1 = \frac{r_0}{\|r_0\|}$.
- For $k = 1, 2, \ldots$, compute q_{k+1} and $h_{i,k}$, $i = 1, 2, \ldots, k+1$, by using the Arnoldi process.
- Form $x_k = x_0 + Q_k y_k$, where y_k is the solution of the least-squares problem $\min_y \|\beta\xi - H_{k+1,k} y\|$.

The least-squares problem $\min_y \|\beta\xi - H_{k+1,k} y\|$ can be solved by the QR factorization. For practical realizations of this QR factorization based on Givens rotations and Householder reflections, we refer to [34, 41, 33].

Alternatively, we can give another procedure to determine an iterate x_k, an approximate to the exact solution x_* of the linear system (1), by using the Krylov subspace $\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$. Recall that

$$r_k \in r_0 - A\mathcal{K}_k(A, r_0).$$

By letting

$$K_k = (r_0, Ar_0, \ldots, A^{k-1} r_0)$$

we have

$$AK_k = (Ar_0, A^2 r_0, \ldots, A^k r_0).$$

Note that the space of the columns of this matrix is $A\mathcal{K}_k$. Now we can find a vector $c \in \mathbb{R}^k$ such that $\|r_0 - AK_k c\|$ is minimized. The vector c could be computed by means of the QR factorization of the matrix AK_k. Once c is available, we would set $x_k = x_0 + K_k c$. Unfortunately, this procedure might be numerically unstable and it constructs a factor R that is not needed.

4 Preconditioning

If it turns out that the Krylov subspace does not contain a good approximate solution for any moderate size k, or if such an approximate solution cannot be easily computed, then one might consider modifying the original linear system (1) to obtain a better Krylov subspace; see [1, 25, 30, 33]. This may be reached by using a preconditioner, say, M. Then the solution x_* can be obtained through effectively solving the modified (or preconditioned) linear system

$$M^{-1}Ax = M^{-1}b.$$

It is widely recognized that preconditioning is the most critical ingredient in the development of efficient solvers for challenging problems in scientific computation, and that the importance of preconditioning is destined to increase even further [1, 25, 15, 24]. Below we enumerate several typical preconditioning methods in order.

(a) Incomplete LU and QR factorizations, which include such kinds of factorizations based on level-of-fill, drop-to-tolerance, and dual threshold, as well as their modified, block and parallel variants [1, 7, 15, 33, 14].

(b) Sparse approximate inverses, which include such kinds of preconditioning strategies based on Frobenius norm minimization and incomplete biconjugation in componentwise and blockwise forms [31, 26, 15].

(c) Matrix splitting iterations, which include the classical Jacobi, Gauss-Seidel, and SOR, as well as their extrapolated and modified variants; and the modern *Hermitian and skew-Hermitian splitting* (**HSS**), *normal and skew-Hermitian splitting* (**NSS**), and the *positive-definite and skew-Hermitian splitting* (**PSS**) iteration methods, as well as their preconditioned and accelerated variants. For more details, we refer to [1, 9, 8, 10, 12] and the references therein.

(d) Algebraic multilevel and multigrid iteration techniques, as well as their parallel versions [2, 3, 42, 4, 6, 15].

Indeed, Krylov subspace iteration methods, when appropriately incorporated with such effective preconditioners, can often rapidly, accurately and robustly solve large sparse linear systems arising from real-world applications; see, e.g., [24, 32, 22, 29, 37, 17].

5 Concluding Remarks

We have provided a brief but concise overview of some of the most promising and typical direct and iterative methods for solving large sparse linear systems, including preconditioning techniques for the Krylov subspace iteration methods, and pointed out possible connections between linear solvers and matrix preconditioners. As have been shown, preconditioning is usually vital to ensure rapid, accurate and stable convergence of Krylov subspace iteration methods, and has been a more active research area than either direct solution methods or Krylov subspace methods, though much effort has already been put in the development of effective preconditioners. Of course, matrix factorizations and matrix splittings provide feasible ways for constructing high-quality and economical preconditioners [1, 13, 15, 20, 16].

References

[1] O. Axelsson, Iterative Solution Methods, *Cambridge University Press*, Cambridge, 1994.

[2] O. Axelsson and P.S. Vassilevski, Algebraic multilevel preconditioning methods. I, *Numer. Math.*, 56 (1989), 157-177.

[3] Z.-Z. Bai, A class of hybrid algebraic multilevel preconditioning methods, *Appl. Numer. Math.*, 19 (1996), 389-399.

[4] Z.-Z. Bai, Parallel hybrid algebraic multilevel iterative methods, *Linear Algebra Appl.*, 267 (1997), 281-315.

[5] Z.-Z. Bai, Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems, *Appl. Math. Comput.*, 109 (2000), 273-285.

[6] Z.-Z. Bai and O. Axelsson, A unified framework for the construction of various algebraic multilevel preconditioning methods, *Acta Math. Appl. Sinica*, 15 (1999), 132-143.

[7] Z.-Z. Bai, I.S. Duff and A.J. Wathen, A class of incomplete orthogonal factorization methods. I: Methods and theories, *BIT Numer. Math.*, 41 (2001), 53-70.

[8] Z.-Z. Bai, G.H. Golub, L.-Z. Lu and J.-F. Yin, Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, *SIAM J. Sci. Comput.*, 26 (2005), 844-863.

[9] Z.-Z. Bai, G.H. Golub and M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM J. Matrix. Anal. Appl.*, 24 (2003), 603-626.

[10] Z.-Z. Bai, G.H. Golub and M.K. Ng, On successive overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, *Numer. Linear Algebra Appl.*, 14 (2007), 319-335.

[11] Z.-Z. Bai and T.-Z. Huang, On the convergence of the relaxation methods for positive definite linear systems, *J. Comput. Math.*, 16 (1998), 527-538.

[12] Z.-Z. Bai and M.K. Ng, Erratum, *Numer. Linear Algebra Appl.*, 19 (2012), 891.

[13] Z.-Z. Bai, J.-C. Sun and D.-R. Wang, A unified framework for the construction of various matrix multisplitting iterative methods for large sparse system of linear equations, *Computers Math. Appl.*, 32 (1996), 51-76.

[14] Z.-Z. Bai and J.-F. Yin, Modified incomplete orthogonal factorization methods using Givens rotations, *Computing*, 86 (2009), 53-69.

[15] M. Benzi, Preconditioning techniques for large linear systems: A survey , *J. Comput. Phy.*, 182 (2002), 418-477.

[16] M. Benzi and A.J. Wathen, Some preconditioning techniques for saddle point problems, *Model Order Reduction: Theory, Research Aspects and Applications*, W. Schilders, H.A. van der Vorst and J. Rommes, eds., Springer-Verlag (Series: Mathematics in Industry), 2008, pp. 195-211.

[17] J. Bosch, D. Kay, M. Stoll and A.J. Wathen, Fast solvers for Cahn-Hilliard inpainting, *SIAM J. Imaging Sci.*, 7 (2014), 67-97.

[18] L. Cvetković and V. Kostić, New subclasses of block H-matrices with applications to parallel decomposition-type relaxation methods, *Numer. Algorithms*, 42 (2006), 325-334.

[19] L. Cvetković and V. Kostić, A note on the convergence of the AOR method, *Appl. Math. Comput.*, 194 (2007), 394-399.

[20] I.S. Duff, The design and use of a sparse direct solver for skew symmetric matrices, *J. Comput. Appl. Math.*, 226 (2009), 50-54.

[21] I.S. Duff, A.M. Erisman and J.K. Reid, Direct Methods for Sparse Matrices, Second Edition, *Oxford University Press*, New York, 1989.

[22] I.S. Duff, S. Gratton, X. Pinel and X. Vasseur, Multigrid based preconditioners for the numerical solution of two-dimensional heterogeneous problems in geophysics, *Int. J. Comput. Math.*, 84 (2007), 1167-1181.

[23] I.S. Duff and S. Pralet, Strategies for scaling and pivoting for sparse symmetric indefinite problems, *SIAM J. Matrix Anal. Appl.*, 27 (2005), 313-340.

[24] H.C. Elman, D.J. Silvester and A.J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, *Oxford University Press*, New York, 2005.

[25] G.H. Golub and C.F. Van Loan, Matrix Computations, Third Edition, *The Johns Hopkins University Press,* Baltimore, 1996.

[26] M. Grote and T. Huckle, Parallel preconditioning with sparse approximate inverses, *SIAM J. Sci. Comput.*, 18 (1997), 838-853.

[27] A. Hadjidimos, Successive overrelaxation (SOR) and related methods, *J. Comput. Appl. Math.*, 123 (2000), 177-199.

[28] A. Hadjidimos, A. Psimarni and A.K. Yeyios, On the convergence of some generalized iterative methods, *Linear Algebra Appl.*, 75 (1986), 117-132.

[29] Z.-K. Huang and K.-W. Chau, A new image thresholding method based on Gaussian mixture model, *Appl. Math. Comput.*, 205 (2008), 899-907.

[30] I.C.F. Ipsen and C.D. Meyer, The idea behind Krylov methods, *American Math. Monthly*, 105 (1998), 889-899.

[31] L.Yu. Kolotilina and A.Yu. Yeremin, Factorized sparse approximate inverse preconditioning I. Theory, *SIAM J. Matrix Anal. Appl.*, 14 (1993), 45-58.

[32] J.-Y. Lin, C.-T. Cheng, Y.-G. Sun and K.-W. Chau, Long-term prediction of discharges in manwan reservoir using artificial neural network models, *Lecture Notes in Computer Science*, 3498 (2005), 1040-1045.

[33] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition, *SIAM*, Philadelphia, PA, 2003.

[34] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, 7 (1986), 856-869.

[35] J.A. Sifuentes, M. Embree and R.B. Morgan, GMRES convergence for perturbed coefficient matrices, with application to approximate deflation preconditioning, *SIAM J. Matrix Anal. Appl.*, 34 (2013), 1066-1088.

[36] G.W. Stewart, Matrix Algorithms, Vol. I: Basic Decompositions, *SIAM*, Philadelphia, PA, 1998.

[37] R. Taormina, K.-W. Chau and R. Sethi, Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon, *Engrg. Appl. Artif. Intell.*, 25 (2012), 1670-1676.

[38] D. Titley-Peloquin, J. Pestana and A.J. Wathen, GMRES convergence bounds that depend on the right-hand-side vector, *IMA J. Numer. Anal.*, 34 (2014), 462-479.

[39] L.N. Trefethen and D. Bau, Numerical Linear Algebra, *SIAM*, Philadelphia, PA, 1997.

[40] R.S. Varga, Matrix Iterative Analysis, *Prentice Hall,* Englewood Cliffs, N.J., 1962.

[41] H.F. Walker, Implementation of the GMRES method using Householder transformations, *SIAM J. Sci. Statist. Comput.*, 9 (1988), 152-163.

[42] D.-R. Wang and Z.-Z. Bai, Parallel multilevel iterative methods, *Linear Algebra Appl.*, 250 (1997), 317-347.

[43] D.M. Young, Iterative Solution of Large Linear Systems, *Academic Press,* New York, 1971.