Method	Inner Product	SAXPY	Matrix-Vector Product	Precond Solve
JACOBI			1^a	
GS		1	1^a	
SOR		1	1^a	
CG	2	3	1	1
GMRES	$i+1$	$i+1$	1	1
BiCG	2	5	1/1	1/1
QMR	2	$8{+}4^{bc}$	1/1	1/1
CGS	2	6	2	2
Bi-CGSTAB	4	6	2	2
CHEBYSHEV		2	1	1

Table 2.1: Summary of Operations for Iteration i. "a/b" means "a" multiplications with the matrix and "b" with its transpose.

[a] This method performs no real matrix vector product or preconditioner solve, but the number of operations is equivalent to a matrix-vector multiply.

[b] True SAXPY operations + vector scalings.

[c] Less for implementations that do not recursively update the residual.

- Computational kernels. Different methods involve different kernels, and depending on the problem or target computer architecture this may rule out certain methods.

Table 2.2 lists the storage required for each method (without preconditioning). Note that we are not including the storage for the original system $Ax = b$ and we ignore scalar storage.

1. Jacobi Method
    - Extremely easy to use, but unless the matrix is "strongly" diagonally dominant, this method is probably best only considered as an introduction to iterative methods or as a preconditioner in a nonstationary method.
    - Trivial to parallelize.
2. Gauss-Seidel Method
    - Typically faster convergence than Jacobi, but in general not competitive with the nonstationary methods.
    - Applicable to strictly diagonally dominant, or symmetric positive definite matrices.
    - Parallelization properties depend on structure of the coefficient matrix. Different orderings of the unknowns have different degrees of parallelism; multi-color orderings may give almost full parallelism.
    - This is a special case of the SOR method, obtained by choosing $\omega = 1$.
3. Successive Over-Relaxation (SOR)

Method	Storage Reqmts
JACOBI	matrix $+ 3n$
SOR	matrix $+ 2n$
CG	matrix $+ 6n$
GMRES	matrix $+ (i+5)n$
BiCG	matrix $+ 10n$
CGS	matrix $+ 11n$
Bi-CGSTAB	matrix $+ 10n$
QMR	matrix $+ 16n^c$
CHEBYSHEV	matrix $+ 5n$

Table 2.2: Storage Requirements for the Methods in iteration i: n denotes the order of the matrix.

[c] Less for implementations that do not recursively update the residual.

- Accelerates convergence of Gauss-Seidel ($\omega > 1$, *over*-relaxation); may yield convergence when Gauss-Seidel fails ($0 < \omega < 1$, *under*-relaxation).
- Speed of convergence depends critically on ω; the optimal value for ω may be estimated from the spectral radius of the Jacobi iteration matrix under certain conditions.
- Parallelization properties are the same as those of the Gauss-Seidel method.

4. Conjugate Gradient (CG)
   - Applicable to symmetric positive definite systems.
   - Speed of convergence depends on the condition number; if extremal eigenvalues are well-separated then superlinear convergence behavior can result.
   - Inner products act as synchronization points in a parallel environment.
   - Further parallel properties are largely independent of the coefficient matrix, but depend strongly on the structure the preconditioner.

5. Generalized Minimal Residual (GMRES)
   - Applicable to nonsymmetric matrices.
   - GMRES leads to the smallest residual for a fixed number of iteration steps, but these steps become increasingly expensive.
   - In order to limit the increasing storage requirements and work per iteration step, restarting is necessary. When to do so depends on A and the right-hand side; it requires skill and experience.
   - GMRES requires only matrix-vector products with the coefficient matrix.
   - The number of inner products grows linearly with the iteration number, up to the restart point. In an implementation based on a simple Gram-Schmidt process the inner products are independent, so together they imply only one synchronization point. A more stable implementation based on modified Gram-Schmidt orthogonalization has one synchronization point per inner product.

6. Biconjugate Gradient (BiCG)
   - Applicable to nonsymmetric matrices.
   - Requires matrix-vector products with the coefficient matrix and its transpose. This disqualifies the method for cases where the matrix is only given implicitly as an operator, since usually no corresponding transpose operator is available in such cases.
   - Parallelization properties are similar to those for CG; the two matrix vector products (as well as the preconditioning steps) are independent, so they can be done in parallel, or their communication stages can be packaged.
7. Quasi-Minimal Residual (QMR)
   - Applicable to nonsymmetric matrices.
   - Designed to avoid the irregular convergence behavior of BiCG, it avoids one of the two breakdown situations of BiCG.
   - If BiCG makes significant progress in one iteration step, then QMR delivers about the same result at the same step. But when BiCG temporarily stagnates or diverges, QMR may still further reduce the residual, albeit very slowly.
   - Computational costs per iteration are similar to BiCG, but slightly higher. The method requires the transpose matrix-vector product.
   - Parallelization properties are as for BiCG.
8. Conjugate Gradient Squared (CGS)
   - Applicable to nonsymmetric matrices.
   - Converges (diverges) typically about twice as fast as BiCG.
   - Convergence behavior is often quite irregular, which may lead to a loss of accuracy in the updated residual.
   - Computational costs per iteration are similar to BiCG, but the method doesn't require the transpose matrix.
   - Unlike BiCG, the two matrix-vector products are not independent, so the number of synchronization points in a parallel environment is larger.
9. Biconjugate Gradient Stabilized (Bi-CGSTAB)
   - Applicable to nonsymmetric matrices.
   - Computational costs per iteration are similar to BiCG and CGS, but the method doesn't require the transpose matrix.
   - An alternative for CGS that avoids the irregular convergence patterns of CGS while maintaining about the same speed of convergence; as a result we often observe less loss of accuracy in the updated residual.
10. Chebyshev Iteration
    - Applicable to nonsymmetric matrices (but presented in this book only for the symmetric case).

- This method requires some explicit knowledge of the spectrum (or field of values); in the symmetric case the iteration parameters are easily obtained from the two extremal eigenvalues, which can be estimated either directly from the matrix, or from applying a few iterations of the Conjugate Gradient Method.
- The computational structure is similar to that of CG, but there are no synchronization points.
- The Adaptive Chebyshev method can be used in combination with methods as CG or GMRES, to continue the iteration once suitable bounds on the spectrum have been obtained from these methods.

Selecting the "best" method for a given class of problems is largely a matter of trial and error. It also depends on how much storage one has available (GMRES), on the availability of A^T (BiCG and QMR), and on how expensive the matrix vector products (and Solve steps with M) are in comparison to `SAXPY`s and inner products. If these matrix vector products are relatively expensive, and if sufficient storage is available then it may be attractive to use GMRES and delay restarting as much as possible.

Table 2.1 shows the type of operations performed per iteration. Based on the particular problem or data structure, the user may observe that a particular operation could be performed more efficiently.

2.5 A short history of Krylov methods[1]

Methods based on orthogonalization were developed by a number of authors in the early '50s. Lanczos' method [142] was based on two mutually orthogonal vector sequences, and his motivation came from eigenvalue problems. In that context, the most prominent feature of the method is that it reduces the original matrix to tridiagonal form. Lanczos later applied his method to solving linear systems, in particular symmetric ones [143]. An important property for proving convergence of the method when solving linear systems is that the iterates are related to the initial residual by multiplication with a polynomial in the coefficient matrix.

The joint paper by Hestenes and Stiefel [122], after their independent discovery of the same method, is the classical description of the conjugate gradient method for solving linear systems. Although error-reduction properties are proved, and experiments showing premature convergence are reported, the conjugate gradient method is presented here as a direct method, rather than an iterative method.

This Hestenes/Stiefel method is closely related to a reduction of the Lanczos method to symmetric matrices, reducing the two mutually orthogonal sequences to one orthogonal sequence, but there is an important algorithmic difference. Whereas Lanczos used three-term recurrences, the method by Hestenes and Stiefel uses coupled two-term recurrences. By combining the two two-term recurrences (eliminating the "search directions") the Lanczos method is obtained.

A paper by Arnoldi [6] further discusses the Lanczos biorthogonalization method, but it also presents a new method, combining features of the Lanczos and

[1] For a more detailed account of the early history of CG methods, we refer the reader to Golub and O'Leary [108] and Hestenes [123].