SIAM J. NUMER. ANAL.
Vol. 27, No. 6, pp. 1542–1568, December 1990

012

A TAXONOMY FOR CONJUGATE GRADIENT METHODS*

STEVEN F. ASHBY†, THOMAS A. MANTEUFFEL‡, AND PAUL E. SAYLOR§

Abstract. The conjugate gradient method of Hestenes and Stiefel is an effective method for solving large, sparse Hermitian positive definite (hpd) systems of linear equations, $Ax = b$. Generalizations to non-hpd matrices have long been sought. The recent theory of Faber and Manteuffel gives necessary and sufficient conditions for the existence of a CG method. This paper uses these conditions to develop and organize such methods. It is shown that any CG method for $Ax = b$ is characterized by an hpd inner product matrix B and a left preconditioning matrix C. At each step the method minimizes the B-norm of the error over a Krylov subspace. This characterization is then used to classify known and new methods. Finally, it is shown how eigenvalue estimates may be obtained from the iteration parameters, generalizing the well-known connection between CG and Lanczos. Such estimates allow implementation of a stopping criterion based more nearly on the true error.

Key words. conjugate gradient methods, Lanczos methods, B-normal matrices

AMS(MOS) subject classifications. 65F10

1. Introduction.

The need to solve an $n \times n$ nonsingular system of linear equations, $Ax = b$, arises in many scientific applications. The matrix A is usually real, but since complex matrices do occur, for example, in computational electromagnetics, we will consider the more general case. If A is large and sparse, direct methods are too expensive and an iterative method is required. When A is Hermitian positive definite (hpd), Hestenes's and Stiefel's method of conjugate gradients (CGHS) is popular and effective, especially when combined with a preconditioner. Its popularity is due in part to optimality: at each step the A-norm of the error is minimized over some subspace. Moreover, CGHS requires no a priori parameter estimates and may be implemented via a 3-term recursion. By minimizing in other than the A-norm, different CG methods result, some of which are applicable to non-hpd matrices. For example, the method of conjugate residuals minimizes the Euclidean norm of the residual at each step and converges for any Hermitian matrix.

Following the theory developed in [16] and [17], we define a CG method to be a gradient method in which the iterates are chosen from a nested sequence of translated Krylov subspaces in such a way that the error is minimal in the given inner product norm at each step. The Krylov subspaces are generated by a preconditioned system matrix, CA. The choice of inner product matrix B and preconditioning matrix C completely determine the sequence of iterates. Although such a sequence exists for any B and C, it may be uncomputable without knowledge of the solution. We thus restrict our attention to those B and C that yield computable methods.

Although computable, a method may require the storage of all past direction vectors unless certain conditions are met. In [16] necessary and sufficient conditions are given for the existence of a 3-term CG method that is optimal in the B-norm, $\|\cdot\|_B = \langle B\cdot, \cdot\rangle^{1/2}$. (Such methods require storing only the past two direction vectors.) In the present paper, these conditions are used to develop and organize CG methods. In particular, we show that *any* CG method for $Ax = b$ is uniquely characterized by an

* Received by the editors April 14, 1988; accepted for publication (in revised form) December 21, 1989. This work was supported in part by National Science Foundation grants DMS-8704169 and DMS-8703226; by Air Force Office of Scientific Research contract AFOSR-86-0061; and by Department of Energy contracts W-7405-ENG-48 and DEFG02-87ER25026.

† Computing & Mathematics Research Division, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, California 94550.

‡ Computational Math Group, University of Colorado, 1200 Larimer St., Denver, Colorado 80204.

§ Department of Computer Science, University of Illinois, 1304 W. Springfield Avenue, Urbana, Illinois 61801.

hpd inner product matrix B and a *left* preconditioning matrix C. This characterization is then used to classify a variety of methods. For each method we also determine the class of matrices for which the 3-term recursion is applicable. We call this class of matrices the *domain of applicability* for the given method. The resulting taxonomy not only imposes order, but also provides insight into how still other methods may be devised. Moreover, by exploring the domain of applicability of each method, new applications for several known methods are revealed.

To aid in the design and classification of CG methods, we identify seven basic CG *patterns*. Specifically, we identify patterns for the B and C matrices. A *method* is an instance of a pattern, that is, a specific choice for B and C. The patterns form the basis for the taxonomy presented in this paper. By examining these patterns, the relationship between methods that seem prima facie unrelated can more readily be seen. Moreover, once a pattern is identified, it may be used to devise new methods and to understand known methods more fully. For example, one of our patterns leads to a new class of methods that is based on commutativity.

This paper also considers the efficient implementation of a CG method. We call such an implementation an *algorithm*; it is the sequence of arithmetic steps used to implement the method. Three algorithms are examined. The first, Orthodir, is the most general, converging to $x = A^{-1}b$ whenever the underlying method is optimal. Two variants, Orthomin and Orthores, are also considered. We next show how eigenvalue estimates for CA may be computed from a tridiagonal matrix constructed from the CG iteration parameters, generalizing the well-known connection between the CGHS and Lanczos methods. Such estimates allow implementation of a stopping criterion based more nearly on the true error, rather than the residual error.

Finally, a word on terminology. Historically [31], CG methods have been called *gradient* methods because the residual is the gradient of the error functional $\frac{1}{2}\langle Ae, e\rangle$ when A is hpd. In this paper, we employ the more general error functional $\frac{1}{2}\langle Be, e\rangle$, whose gradient is Be. Another term the reader may be familiar with is *polynomial* method, which has its origin in the fact that the error e is a polynomial in CA times the initial error.

1.1. Related work. An early attempt at taxonomy is given by Hestenes in [20]. He observed that many CG methods are equivalent to CGHS on some preconditioned problem, $\tilde{A}\tilde{x} = \tilde{b}$. Unfortunately, this approach does not reveal the full domain of applicability of the resulting method. For example, it leads to the erroneous conclusion that the preconditioner in PCG must be hpd. As we will show, it only needs to be Hermitian.

The taxonomy most similar to ours is that of Joubert and Young [27]. This is not surprising since they also base their work on [16]. However, their approach is different than ours. In particular, they use an "auxiliary" matrix Z. (In our notation, $B = ZA$.) They also focus much more on normality and definiteness.

Dennis and Turner provide a taxonomy for descent methods in [12]. There the linear system $Ax = b$ is solved by finding the minimum of a quadratic functional. The iterates are again chosen from a nested sequence of translated subspaces, but they are not necessarily Krylov subspaces. Since each new space is required to produce a descent direction, convergence is guaranteed if the amount of descent can be bounded away from zero. By requiring a descent direction at each step, however, the application of these methods to a large class of indefinite and nonself-adjoint systems is precluded. We explore methods for such systems in § 5.4.

When B is definite, but not Hermitian, $\langle B\cdot, \cdot\rangle$ is a definite sesquilinear form,

which may be used to define a one-sided orthogonality property. This is the basis for the Orthogonal Error Methods discussed in [17]. Many of our results generalize to OEM's in a straightforward manner. However, the estimation of eigenvalues is more complicated; and since $\langle B\cdot,\cdot\rangle$ no longer defines a norm, some optimality properties may be lost. In the present work we examine only CG methods; OEM's will be addressed in a future report.

1.2. Outline. We review some basic theory in § 2. The algorithm Orthodir is discussed in § 3; Orthomin and Orthores are examined in § 4. In § 5 we introduce our taxonomy. We first identify and describe seven basic CG patterns (§ 5.1) and show how ten well-known methods fit into the taxonomy (§ 5.2). The role of commutativity in CG methods is considered in § 5.3; the domain of applicability for each pattern is explored in § 5.4. In § 6 the relationship between the CGHS and Lanczos methods is generalized and stopping criteria are discussed. We summarize our results in § 7.

2. Basic theory. In this section some basic theory of conjugate gradient methods is reviewed. It is first noted that a CG method is a gradient (or polynomial) method [31] that minimizes the error in an inner product norm at each step. The class of matrices for which an "economical" method exists, that is, for which only a few previous direction vectors are needed, is then presented. Except for certain anomalous cases, this class consists of diagonalizable matrices whose eigenvalues lie on a line segment in the complex plane.

2.1. Gradient methods. Given an initial guess x_0, a gradient method produces a sequence of approximations to $x = A^{-1}b$ by

$$x_{i+1} = x_i + d_i \tag{2.1}$$

where d_i is an element of the Krylov space of dimension at most $i+1$ generated by r_0 and A, $V_{i+1}(r_0, A)$. This Krylov space is defined by

$$V_{i+1}(r_0, A) = \mathrm{sp}\{r_0, Ar_0, A^2r_0, \cdots, A^ir_0\},$$

the space spanned by the enclosed vectors. One gradient method differs from another only in the way d_i is chosen from V_{i+1}. (Symbols r_0 and A will be omitted unless necessary for clarity.)

2.2. CG methods. Let $\langle\cdot,\cdot\rangle$ denote the usual Euclidean inner product. If B is an hpd matrix, then $\langle B\cdot,\cdot\rangle$ defines an inner product, and B is called an *inner product matrix*. Next let $e_i = x - x_i$ be the error at step i; equation (2.1) gives

$$e_{i+1} = e_i - d_i. \tag{2.2}$$

If $d_i \in V_{i+1}$ is chosen to minimize $\|e_{i+1}\|_B \equiv \langle Be_{i+1}, e_{i+1}\rangle^{1/2}$, then a conjugate gradient method results. We denote this method by CG(B, A). Choosing d_i in this way is equivalent [31] to enforcing *B-orthogonality* between e_{i+1} and V_{i+1},

$$\langle Be_{i+1}, z\rangle = 0 \quad \text{for any } z \in V_{i+1},$$

abbreviated by $e_{i+1} \perp_B V_{i+1}$. Since $V_i \subset V_{i+1}$ and $e_i \perp_B V_i$ (from the previous step), we have $d_i \perp_B V_i$. Thus, d_i is the unique (up to scale) vector satisfying both $d_i \in V_{i+1}$ and $d_i \perp_B V_i$.

2.3. A basis for V_{i+1}. To find d_i we first construct a B-orthogonal basis, $\{p_j\}_{j=0}^{i}$, for V_{i+1}:

$$p_{k+1} = Ap_k - \sum_{j=0}^{k} \sigma_{kj} p_j, \quad \sigma_{kj} = \frac{\langle BAp_k, p_j \rangle}{\langle Bp_j, p_j \rangle}, \quad k = 0, 1, \ldots, i-1 \tag{2.3}$$

where $p_0 = r_0 = b - Ax_0$ is the initial residual [16]. The vectors $\{p_j\}$ are called *direction vectors*. By construction, $p_i \in V_{i+1}$ and $p_i \perp_B V_i$. Therefore, $d_i = \alpha_i p_i$ for some scalar α_i. Equation (2.1) then becomes

$$x_{i+1} = x_i + \alpha_i p_i. \tag{2.4}$$

Since $e_{i+1} \perp_B V_{i+1}$, it is easy to determine α_i. The relation

$$0 = \langle Be_{i+1}, p_i \rangle = \langle Be_i, p_i \rangle - \alpha_i \langle Bp_i, p_i \rangle$$

gives

$$\alpha_i = \frac{\langle Be_i, p_i \rangle}{\langle Bp_i, p_i \rangle}. \tag{2.5}$$

The *computability* of α_i is discussed in § 3.2 and § 4.4. By this we mean the evaluation of the right-hand side of (2.5), which involves the unknown error, e_i.

2.4. An economical recursion. The method CG(B, A),

$$x_{i+1} = x_i + \alpha_i p_i,$$
$$p_{i+1} = Ap_i - \sum_{j=0}^{i} \sigma_{ij} p_j$$

yields the solution, $x = A^{-1}b$, in at most n steps for *any* nonsingular matrix A (assuming exact arithmetic). The recursion for p_{i+1} appears to require storing *all* past direction vectors, which is impractical for large problems. Thus, a class of matrices is sought for which (2.3) naturally truncates to an *s-term recursion*:

$$p_{k+1} = Ap_k - \sum_{j=k-s+2}^{k} \sigma_{kj} p_j. \tag{2.6}$$

Only s direction vectors are required. If s is "small," the recursion is said to be *economical* and the corresponding method is called *s-term* CG(B, A).

To proceed further, a few definitions are needed. Let $d(A)$ be the degree of the minimum polynomial of A, and let $d(r_0, A)$ be the degree of the minimum polynomial for r_0 with respect to A. Next, define the *B-adjoint* of A to be the unique matrix A^+ satisfying

$$\langle BAx, y \rangle = \langle Bx, A^+ y \rangle$$

for every x and y. Note that

$$A^+ = (BAB^{-1})^* = B^{-1}A^*B \tag{2.7}$$

where A^* is the adjoint of A in the Euclidean inner product, namely, $A^* = \bar{A}^T$. The matrix A is said to be *B-normal* if and only if $AA^+ = A^+A$. This is true if and only if A^+ is a polynomial (of some degree) in the matrix A [16]. If η is the smallest degree for which this is true, then η is the *normal degree of A*. We say A is *B-normal(s)* if A is B-normal with $\eta \leq s$. Those matrices for which (2.6) is correct may now be characterized as follows.

THEOREM 2.1. *The s-term CG(B, A) method yields the exact solution of $Ax = b$ in at most $d(r_0, A)$ steps for every x_0 if and only if $d(A) \leq s$ or A is B-normal$(s-2)$.*

Proof. See [16]. □

Remark. A similar result holds for singular A, but we confine ourselves to nonsingular A for convenience.

If A does not satisfy the conditions of Theorem 2.1, the recursion (2.6) cannot, in general, be used to generate B-orthogonal direction vectors. Consequently, optimality in the B-norm is not assured, so we do not have a CG method. If the method is employed anyway, it is called a *truncated* CG method (§ 5.5).

2.5. B-normal(1) matrices. In [16] it is shown that if A is B-normal(η) for $\eta > 1$, then $d(A) \leq \eta^2$. Since matrices with at most η^2 distinct eigenvalues are of little interest, we shall assume $\eta \leq 1$ in the remainder of this paper. This corresponds to $s = 3$. Henceforth CG(B, A) will refer to the 3-term CG method that minimizes $\|e_{i+1}\|_B$ over $V_{i+1}(r_0, A)$. By Theorem 2.1 this method is optimal in the B-norm for every x_0 if and only if $d(A) \leq 3$ or A is B-normal(1). The anomalous case $d(A) \leq 3$ is excluded, and we instead focus our attention on B-normal(1) matrices. To better understand CG methods, a characterization of such matrices is useful and will aid development of the taxonomy in § 5.

The matrix A is B-normal(1) if and only if A has the form [16], [27]

$$A = e^{i\theta}\left(i\frac{r}{2}I + G\right), \quad r \geq 0, \quad 0 \leq \theta \leq 2\pi, \quad G^+ = G, \tag{2.8}$$

where G^+ is the B-adjoint of G and $i = \sqrt{-1}$. Characterization (2.8) implies that A is the translation and rotation of a B-self-adjoint matrix ($G^+ = G$). An important subclass arises when $r = 0$ and $\theta = 0$. Then the matrix A is B-self-adjoint; that is, $A^+ = B^{-1}A^*B = A$, which holds if and only if BA is Hermitian.

Given A we might ask if there exists an hpd B such that A is B-normal(1). The answer [16], [27] is: A is B-normal(1) for some B if and only if A is similar to a diagonal matrix and the convex hull of the spectrum of A is a line segment in the complex plane. Moreover, A is B-self-adjoint for some B if and only if A is similar to a Hermitian matrix (and so has real eigenvalues). In other words, a 3-term CG method exists for A if and only if A is diagonalizable and has collinear eigenvalues (or $d(A) \leq 3$). The matrix B may be difficult to find. In theory, the similarity transformation to diagonal form yields a B. In practice, however, it is necessary to search for a B for which (2.8) is satisfied. In § 5 a variety of B are discussed for various A.

In the next two sections we present algorithms that implement CG(B, A). The matrix A is assumed to be B-normal(1).

3. The algorithm Orthodir. Assume A is B-normal(1) for a given inner product matrix B. Then the following algorithm, which we call Orthodir(B, A), implements CG(B, A):

$$p_0 = r_0 \tag{3.1a}$$

$$\alpha_i = \frac{\langle Be_i, p_i\rangle}{\langle Bp_i, p_i\rangle} \tag{3.1b}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{3.1c}$$

$$r_{i+1} = r_i - \alpha_i Ap_i \tag{3.1d}$$

$$\gamma_i = \frac{\langle BAp_i, p_i\rangle}{\langle Bp_i, p_i\rangle} \tag{3.1e}$$

$$\sigma_i = \frac{\langle BAp_i, p_{i-1}\rangle}{\langle Bp_{i-1}, p_{i-1}\rangle} \tag{3.1f}$$

$$p_{i+1} = Ap_i - \gamma_i p_i - \sigma_i p_{i-1} \tag{3.1g}$$

where x_i is the current approximation to $x = A^{-1}b$, $r_i = b - Ax_i$ is the residual, and p_i is the current direction vector.

Readers familiar with the "classical" CG algorithm of Hestenes and Stiefel may find Orthodir to be strange. However, it is, in general, the most robust implementation of CG(B, A). In the next section, two alternative algorithms, Orthomin and Orthores, are presented.[1]

3.1. Preconditioning. To solve $Ax = b$ we need not restrict ourselves to this system. The solution x also may be found from the *preconditioned problem* $QAP\tilde{x} = Qb$, where Q and P are nonsingular linear transformations and $P\tilde{x} = x$. Although P and Q usually represent the inverse of an incomplete factorization, they might also represent more complicated procedures, say, a partial multigrid sweep or single step of SSOR. Ideally, $\tilde{A} = QAP$ has a smaller condition number than A, but not necessarily. For if A is arbitrary, and we wish to use a 3-term CG method, Q and P must be chosen so that $\tilde{A}$ is $\tilde{B}$-normal(1) for some $\tilde{B}$, even if the condition number of $\tilde{A}$ is larger than that of A. The normal equations, $A^*Ax = A^*b$, exemplify this.

To derive a preconditioned algorithm consider Orthodir$(\tilde{B}, \tilde{A})$. This iteration converges to $\tilde{x}$ in at most $d(Qr_0, QAP)$ steps; the desired solution is $x = P\tilde{x}$. The algorithm is easily rearranged, however, to give x directly, as will now be shown.

Let $\tilde{e}_i$, $\tilde{x}_i$, $\tilde{r}_i$, and $\tilde{p}_i$ be the vectors generated by Orthodir$(\tilde{B}, \tilde{A})$. If e_i, x_i, r_i, and p_i are the corresponding vectors for A, then $e_i = P\tilde{e}_i$, $x_i = P\tilde{x}_i$, $r_i = Q^{-1}\tilde{r}_i$, and $p_i = P\tilde{p}_i$. Simple algebra gives the following algorithm, which we call Odir(B, C, A):

$$p_0 = Cr_0 \tag{3.2a}$$

$$\alpha_i = \frac{\langle Be_i, p_i\rangle}{\langle Bp_i, p_i\rangle} \tag{3.2b}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{3.2c}$$

$$r_{i+1} = r_i - \alpha_i Ap_i \tag{3.2d}$$

$$\gamma_i = \frac{\langle BCAp_i, p_i\rangle}{\langle Bp_i, p_i\rangle} \tag{3.2e}$$

[1] These names, Orthodir, Orthomin, and Orthores, have been used by Young et al. [23], [27], [35]. For brevity we often refer to them as Odir, Omin, and Ores, respectively.

$$\sigma_i = \frac{\langle BCAp_i, p_{i-1}\rangle}{\langle Bp_{i-1}, p_{i-1}\rangle} \tag{3.2f}$$

$$p_{i+1} = CAp_i - \gamma_i p_i - \sigma_i p_{i-1} \tag{3.2g}$$

where $B = P^{-*}\tilde{B}P^{-1}$ and $C = PQ$. (Note that C is nonsingular.) This is the same algorithm as Orthodir(B, CA). Therefore, only left preconditioning need be considered: Right preconditioning may be effected by incorporating it into the left preconditioner and inner product. This result has two benefits. First, the vectors x_i, e_i, and p_i are unscaled, which simplifies analysis. Second, it simplifies taxonomy. For once the matrices B, C, and A are specified, the CG method is determined; we call this method CG(B, C, A). Moreover, *any* 3-term CG method may be implemented by Odir(B, C, A) for some B, C, and A. The following result is obtained by replacing A with CA in Theorem 2.1.

THEOREM 3.1. *The s-term $CG(B, C, A)$ method yields the exact solution of $Ax = b$ in at most $d(Cr_0, CA)$ steps for every x_0 if and only if $d(CA) \leq s$ or CA is B-normal$(s-2)$.*

One final remark: The conventional way to describe the preconditioned conjugate gradient method is to state that it is the classical CG method of Hestenes and Stiefel, which we call CGHS, applied to the system $C^{1/2}AC^{1/2}C^{-1/2}x = C^{1/2}b$. Algebraic manipulations yield the algorithm above when $B = A$. This approach is lacking in two respects: (1) It requires that both C and A be hpd; and (2) it requires minimization of the unscaled error in the $B = A$ norm. Now consider the problem $QAPP^{-1}x = Qb$. CGHS on this system is equivalent to CG(B, C, A), where $C = PQ$ and $B = P^{-*}QA$. Clearly, B is hpd if and only if QAP is hpd. This more general approach is still lacking in that it requires that BCA be hpd. This requirement is much stronger than necessary; the theory in § 2 requires only that CA be B-normal(1).

3.2. Computability. Since B is hpd, α_i, γ_i, and σ_i are defined. However, the numerator of α_i involves the unknown quantity e_i, the error at step i. Thus, B and A must be chosen so that α_i is *computable,* that is, expressed in terms of known entities. In general, we regard r_i as the basic known quantity; that is, we assume that A and b are the basic irreducible entities. It will be shown in § 4.4 that α_i is computable whenever C^*Be_i is computable.

3.3. Properties of Odir. Odir(B, C, A) is optimal in the B-norm if and only if either CA is B-normal(1) or $d(CA) \leq 3$. The direction vectors are still B-orthogonal, but now $\|e_{i+1}\|_B$ is minimized over $x_0 + V_{i+1}(Cr_0, CA)$. Thus, the preconditioning matrix C affects only the Krylov space and not the error norm. We also have the following result.

THEOREM 3.2. *If B is hpd and CA is B-normal(1), then the vectors generated by Odir(B, C, A) satisfy the following orthogonality relations for $i \leq d(s_0, CA) - 1$:*

(a) $\langle Be_i, p_j\rangle = 0, \quad j < i,$
(b) $\langle Bp_i, p_j\rangle = 0, \quad j \neq i,$
(c) $\langle Be_i, s_j\rangle = 0, \quad j < i,$

where $s_j = Cr_j$ is the jth preconditioned residual.

Proof. Parts (a) and (b) are intrinsic properties of the method; see § 2. For (c) consider $s_j \in \text{sp}\{s_0, \cdots, s_j\} \subset \text{sp}\{p_0, \cdots, p_j\}$. Thus, $s_j = \sum_{k=0}^{j} \tau_k p_k$ and the result follows from (b). □

COROLLARY 3.3. *Under the hypotheses of Theorem* 3.2:

(d) $\langle BCAp_i, p_j\rangle = 0, \quad j \neq i-1, i, i+1,$
(e) $\sigma_j = e^{2i\theta}\langle Bp_j, p_j\rangle / \langle Bp_{j-1}, p_{j-1}\rangle$

where θ is as given in (2.8).

Proof. The first result follows from the 3-term recursion (3.2g) for the p_i and Theorem 3.2b. As in (2.8), if CA is B-normal(1), then $CA = e^{i\theta}(i\frac{r}{2}I + G)$ for some $r \geq 0$, $0 \leq \theta \leq 2\pi$, and $G = G^+$. Taking the B-adjoint of CA and rearranging terms yields $(CA)^+ = e^{-2i\theta}CA - ire^{-i\theta}I$. Therefore, the numerator of σ_j is

$$\langle BCAp_j, p_{j-1}\rangle = \langle Bp_j, (CA)^+ p_{j-1}\rangle = e^{2i\theta}\langle Bp_j, CAp_{j-1}\rangle + ire^{i\theta}\langle Bp_j, p_{j-1}\rangle.$$

Result (e) now follows from (3.2g) and Theorem 3.2b. □

Remark. $e^{2i\theta} = 1$ and $r = 0$ when CA is B-self-adjoint.

Theorem 3.2 lists the basic orthogonality relations of CG(B, C, A). From § 2.2 it follows that e_{i+1} is B-orthogonal to V_{i+1}. By Theorem 3.2c e_{i+1} is also B-orthogonal to the space spanned by the preconditioned residuals. This distinction, between the spaces spanned by $\{p_j\}$ and $\{s_j\}$, is an important one. For although $\text{sp}\{s_0, \cdots, s_i\} \subset V_{i+1}$, equality need *not* hold. To see why, suppose some $\alpha_j = 0$. Then the space spanned by the residuals is unchanged from step j to step $j+1$. Consequently, s_{j+1} cannot be used to generate the new Krylov space. However, if it can be guaranteed that $\alpha_j \neq 0$ for every j, then a more economical form of the algorithm exists. This is the topic of the next section.

4. Orthomin and Orthores. When the preconditioned residuals $\{s_j\}_{j=0}^{i}$ span V_{i+1}, a more efficient implementation of CG(B, C, A) is possible. The resulting algorithm is Omin(B, C, A):

$$\hat{p}_0 = Cr_0 \tag{4.1a}$$

$$\hat{\alpha}_i = \frac{\langle Be_i, \hat{p}_i\rangle}{\langle B\hat{p}_i, \hat{p}_i\rangle} \tag{4.1b}$$

$$x_{i+1} = x_i + \hat{\alpha}_i\hat{p}_i \tag{4.1c}$$

$$r_{i+1} = r_i - \hat{\alpha}_i A\hat{p}_i \tag{4.1d}$$

$$s_{i+1} = Cr_{i+1} \tag{4.1e}$$

$$\beta_i = -\frac{\langle BCAe_{i+1}, \hat{p}_i\rangle}{\langle B\hat{p}_i, \hat{p}_i\rangle} \tag{4.1f}$$

$$\hat{p}_{i+1} = s_{i+1} + \beta_i\hat{p}_i. \tag{4.1g}$$

This algorithm differs from Odir in that it uses s_{i+1} to compute $\hat{p}_{i+1}$, that is, to generate V_{i+2}. This is possible if and only if $\hat{\alpha}_j \neq 0$. If some $\hat{\alpha}_j = 0$, the iteration is trapped in the current Krylov space: $x_{j+k} = x_j$ for $k \geq 0$. Odir avoids this problem by using a 3-term recursion for p_{j+1}, but at greater expense. (Although $\alpha_j = 0$ is possible in Odir, the 3-term recursion (3.2g) always forces the iteration into the next Krylov space.)

Note that Omin(A, I, A) is the classical algorithm of Hestenes and Stiefel when A is hpd.

4.1. Properties of Omin. Omin is more efficient than Odir, requiring less work per step and less storage. If CA is B-normal(1) and each $\hat{\alpha}_j$ is nonzero, then Omin is *equivalent* to Odir: given the same x_0, both algorithms produce the same sequence of iterates, which follows because both algorithms implement the same method, CG(B, C, A). Equations (3.2c) and (4.1c) imply that $\alpha_i p_i = \hat{\alpha}_i\hat{p}_i$. Thus, the orthogonality relations listed in Theorem 3.2 hold for Omin with p_i replaced by $\hat{p}_i$ (provided no $\hat{\alpha}_j = 0$). In addition, equation (4.1g) admits the following useful results.

COROLLARY 4.1. *Assume the hypotheses of Theorem* 3.2. *Then*

(f) $\langle Be_i, \hat{p}_i \rangle = \langle Be_i, s_i \rangle$,
(g) $\langle B\hat{p}_i, \hat{p}_i \rangle = \langle B\hat{p}_i, s_i \rangle$.

If, in addition, BCA is definite, then

(h) $\hat{\alpha}_i \neq 0$,
(i) $\beta_j = e^{2i\theta} \langle Be_{j+1}, s_{j+1} \rangle / \overline{\langle Be_j, s_j \rangle}$,

where θ *is as in* (2.8).

Proof. Parts (f) and (g) follow from (4.1g) and Theorem 3.2. For (h) notice that

$$\hat{\alpha}_i = \frac{\langle Be_i, \hat{p}_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle} = \frac{\langle Be_i, s_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle} = \frac{\langle Be_i, CAe_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle} = \frac{\langle e_i, BCAe_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle} \neq 0$$

since B is hpd and BCA is definite. For part (i) consider the numerator of β_j in (4.1f):

$$\langle BCAe_{j+1}, \hat{p}_j \rangle = \langle Be_{j+1}, (CA)^+ \hat{p}_j \rangle = e^{2i\theta} \langle Be_{j+1}, CA\hat{p}_j \rangle$$

from $(CA)^+ = e^{-2i\theta} CA - ire^{-i\theta} I$ and Theorem 3.2a. Since $\hat{\alpha}_j \neq 0$, (4.1d) and (4.1e) yield $CA\hat{p}_j = \hat{\alpha}_j^{-1}(s_j - s_{j+1})$. Combining this with Theorem 3.2c, equation (4.1b), and Corollary 4.1f gives (i). □

Remark. β_j is real when CA is B-self-adjoint or B-skew-adjoint.

When BCA is definite, Corollary 4.1h guarantees each $\hat{\alpha}_i \neq 0$. Thus, the preconditioned residuals span V_{i+1}, and Omin will converge. On the other hand, if BCA is indefinite, there exists an x_0 such that $\hat{\alpha}_0 = 0$. This yields the following.

THEOREM 4.2. *Omin*(B, C, A) *converges to* $x = A^{-1}b$ *for every* x_0 *if and only if* BCA *is definite and either* CA *is* B*-normal*(1) *or* $d(CA) \leq 2$.

Proof. See the discussion above and [27]. □

If CA is B-self-adjoint and BCA is definite, then BCA is Hermitian definite. In this case, we may assume BCA is hpd without loss of generality.

Since Odir and Omin are equivalent when BCA is definite, the former may be derived from the latter. A derivation will now be given for completeness. The 3-term recursion for the p_i of Odir is obtained as follows from the expression for the $\hat{p}_i$ of Omin:

$$\begin{aligned} \hat{p}_{i+1} &= s_{i+1} + \beta_i \hat{p}_i = s_i - \hat{\alpha}_i CA\hat{p}_i + \beta_i \hat{p}_i \\ &= -\hat{\alpha}_i CA\hat{p}_i + (1 + \beta_i)\hat{p}_i - \beta_{i-1}\hat{p}_{i-1}. \end{aligned} \tag{4.2}$$

Let $\pi_i = \prod_{k=0}^{i} (-\hat{\alpha}_k)$ and $p_{i+1} = 1/\pi_i \hat{p}_{i+1}$. Then

$$p_{i+1} = CAp_i - \left(\frac{1 + \beta_i}{\hat{\alpha}_i} \right) p_i - \left(\frac{\beta_{i-1}}{\hat{\alpha}_i \hat{\alpha}_{i-1}} \right) p_{i-1},$$

which is the recursion of Odir. We may verify

$$\gamma_i = \frac{1 + \beta_i}{\hat{\alpha}_i} \quad \text{and} \quad \sigma_i = \frac{\beta_{i-1}}{\hat{\alpha}_i \hat{\alpha}_{i-1}}. \tag{4.3}$$

Moreover, $\alpha_i p_i = \hat{\alpha}_i \hat{p}_i$, which shows more clearly that the two algorithms generate the same sequence of iterates, $\{x_{i+1}\}$.

Likewise, the $\hat{p}_i$ of Omin may be derived from the p_i of Odir by multiplying (3.2d) by C, substituting this into (3.2g), and then rearranging terms.

4.2. The algorithm Orthores. A 3-term recursion for the iterates (and hence the residuals) gives a third algorithm, which we call Ores(B, C, A):

$$s_0 = Cr_0, \tag{4.4a}$$

$$\mu_i = \frac{\langle Be_i, s_i\rangle\langle Be_{i-1}, s_{i-1}\rangle}{\langle Bs_i, s_i\rangle\langle Be_{i-1}, s_{i-1}\rangle - \langle Bs_i, s_{i-1}\rangle\langle Be_{i-1}, s_i\rangle}, \tag{4.4b}$$

$$\rho_i = \left(1 + \mu_i \frac{\langle Bs_i, s_{i-1}\rangle}{\langle Be_{i-1}, s_{i-1}\rangle}\right)^{-1}, \tag{4.4c}$$

$$x_{i+1} = \rho_i(x_i + \mu_i s_i) + (1 - \rho_i)x_{i-1}, \tag{4.4d}$$

$$r_{i+1} = \rho_i(r_i - \mu_i A s_i) + (1 - \rho_i)r_{i-1}, \tag{4.4e}$$

$$s_{i+1} = Cr_{i+1}. \tag{4.4f}$$

This algorithm, similar to the one given in Table 3 of [23], is algebraically equivalent to Omin, converging if and only if Omin converges. If Omin determines $\hat{\alpha}_j = 0$, then Ores determines $\mu_j = 0$, and conversely. Both algorithms compute $x_{j+1} = x_j$, but make no further progress toward $x = A^{-1}b$: Omin is unable to generate V_{j+2} and Ores is unable to compute ρ_{j+1}.

Ores may be derived from Omin by substituting $\hat{\alpha}_j\hat{p}_j = x_{j+1} - x_j$ into equation (4.1g) to obtain

$$x_{i+1} = \hat{\alpha}_i s_i + \left(1 + \frac{\hat{\alpha}_i\beta_{i-1}}{\hat{\alpha}_{i-1}}\right)x_i - \left(\frac{\hat{\alpha}_i\beta_{i-1}}{\hat{\alpha}_{i-1}}\right)x_{i-1}.$$

We may verify

$$\mu_i = \frac{\hat{\alpha}_i\hat{\alpha}_{i-1}}{\hat{\alpha}_{i-1} + \hat{\alpha}_i\beta_{i-1}} \quad \text{and} \quad \rho_i = 1 + \frac{\hat{\alpha}_i\beta_{i-1}}{\hat{\alpha}_{i-1}}.$$

Note that μ_i simplifies to

$$\mu_i = \frac{\langle Be_i, s_i\rangle}{\langle Bs_i, s_i\rangle}$$

when CA is B-self-adjoint or B-skew-adjoint.

4.3. Related work. Joubert and Young [27] have examined Omin and Ores in detail, modifying the proof in [16] to obtain necessary and sufficient conditions for convergence. Their conditions are equivalent to those given above. (Note that the anomalous case for Omin and Ores is $d(A) \leq 2$.) Young and Jea [35] have considered the use of Omin and Ores on matrices that are not B-normal(1). Despite the loss of optimality, these *truncated* algorithms are useful in many applications. Also see § 5.5.

4.4. Practical considerations. When BCA is definite, Omin and Ores are algebraically equivalent to Odir: All three converge in at most $d(Cr_0, CA)$ steps, which follows because each algorithm implements the same method. Since Omin requires one less vector update and one less inner product per step, as well as less storage, than Odir, Omin should be used whenever possible. Also, since Ores is more expensive than Omin, yet no more general, we will ignore it in the remainder of the paper.

If BCA is indefinite, Omin may still be used, but the previous direction vector, $\hat{p}_{i-1}$, should be stored. Then, if $\hat{\alpha}_i = 0$ (or is nearly zero), control can switch to the

3-term recursion of Odir to get p_{i+1}. This hybrid algorithm has been studied in [6] and [7] for $B = A^2$.

Regardless of the algorithm, matrix-vector multiplications usually require the most expense. Both Odir and Omin seem to require three multiplications, one each for B, C, and A. However, if the inner product matrix B is chosen appropriately, and the computations arranged correctly, this multiplication can be avoided. This is true for the methods we will discuss in § 5.

Many CG methods require both CAp_i and Cr_{i+1}. Since $Cr_{i+1} = Cr_i - \alpha_i CAp_i$, the preconditioned residual is available at the cost of a vector update. Therefore only one C multiplication per step is necessary. Also note that r_i is not always required, and so need not be computed. However, in several methods we need the auxiliary vector $q_i = C^{-1}p_i$, which may be computed recursively by

$$q_{i+1} = Ap_i - \gamma_i q_i - \sigma_i q_{i-1} \tag{4.5}$$

with $q_0 = r_0$ and $q_{-1} = 0$.

Equation (4.5) yields variants of Odir and Omin that are computable whenever C^*Be_i is computable. In addition, when BCA is definite, results 4.1f and 4.1g allow several expressions for $\hat{\alpha}_i$. The appropriate choice depends on B, C, and A. Finally, notice that Corollaries 3.3e and 4.1i may be used to rewrite σ_i and β_i in terms of previously computed inner products. These are significant savings in any implementation of Odir and Omin.

5. A taxonomy for CG methods. Any CG method is characterized by exactly three matrices: an hpd inner product matrix B, a left preconditioning matrix C, and the system matrix A. This characterization is the foundation for our taxonomy, which we develop in this section. We use this taxonomy to classify a variety of methods, to explore their domains of applicability, and to suggest several new methods based on commutativity.

To better understand the taxonomy below, consider how a new 3-term CG method might be devised: Given the nonsingular system $Ax = b$, an hpd B and nonsingular C must be chosen so that

(i) α_i is computable
(ii) CA is B-normal(1).

The computability of α_i has been mentioned in § 3.2 and § 4.4; it is essential for a practical algorithm. Condition (ii) guarantees that each iterate of CG(B, C, A) is optimal in the B-norm over the current Krylov space. Ideally, C approximates A^{-1}, thus improving convergence. However, this is not always the case, as the normal equations illustrate. Finally, when choosing B and C, the expense of the resulting algorithm should be considered.

Our taxonomy is presented in four parts. In § 5.1 we introduce seven basic CG *patterns*. Particular methods are obtained by specifying the matrices that define a pattern. Thus, a CG pattern is simply a generalized CG method. In § 5.2 we discuss ten known methods and show that each is an instance of one or more of these patterns. In § 5.3 we examine the role of commutativity in CG methods. In particular, we examine commutative preconditioners, that is, preconditioning matrices C for which $CA = AC$. The simplest commutative preconditioner is a polynomial in the matrix A. Two other commutativity properties are also considered.

In §§ 5.1–5.3 we assume that CA is B-self-adjoint, an important special case of B-normality. In § 5.4 we describe the full domain of applicability for each of the patterns in § 5.1. (Recall that the domain of applicability for a method is the class of

matrices for which that method is guaranteed to converge to $x = A^{-1}b$ for any x_0.) To determine this domain we state necessary and sufficient conditions under which CA is B-normal(1) for the given pattern. This study not only suggests several new CG methods, but also reveals that several existing methods are applicable to a larger class of matrices than previously known.

5.1. Basic CG patterns. The premise of this paper is that every CG method is an instance of CG(B, C, A) for some B, C, and A. A particular method for solving $Ax = b$ is obtained by specifying a choice for B and C. It is useful, however, to have a level of abstraction between the extremes of CG(B, C, A) and particular methods. Thus, in this section, we introduce CG *patterns*, and these constitute our taxonomy. A pattern gives us a way of obtaining a computable 3-term CG method by relating B, C, and A so that CA is B-normal(1). For example, in the CG method of Hestenes and Stiefel, the pattern is to take the inner product matrix to be the system matrix. By identifying various patterns, we can more readily see the relationship between methods that seem prima facie unrelated. Moreover, once a pattern is identified, it may be used to devise new methods. We remark that the list of patterns below is by no means complete, nor are they mutually exclusive. We have attempted only to identify important and useful patterns.

In Table 5.1 we list seven CG patterns; they may be distinguished by their entries in the B and CA columns. The first four patterns, P1–P4, are motivated by four well-known methods (see below), and may be viewed as generalizations of these methods. In each of these patterns we assume B has the form ZA, where Z is some *computable matrix*. By this we mean that the matrix-vector product $w = Zv$ is computable for any known vector v. Examples of computable matrices include A, A^*, C, and I; the matrix A^{-1} is an example of an uncomputable matrix. Note that $B = ZA$ guarantees a computable method (§ 3.2). Of course, it is not necessary that $B = ZA$ for a computable Z. As remarked in § 4.4, a method is computable if C^*Be_i is computable. This observation is used in patterns P5–P7.

In the column labeled "Odir Restrictions" we list sufficient conditions under which B is hpd and BCA is Hermitian. If these conditions are satisfied, Odir(B, C, A) will converge. If the conditions listed under "Omin Restrictions" are met, B and BCA are both hpd, and so Omin(B, C, A) will converge. In either case the matrix CA is B-self-adjoint. Since these matrices commonly occur in practice and are easy to characterize, we deal with them first. We consider the more general class of B-normal(1) matrices in § 5.4, where the full domain of applicability for each pattern is discussed. Computable expressions for α_i and $\hat{\alpha}_i$ are also given in Table 5.1. To obtain efficient algorithms, in which only one C-matvec and one A-matvec are required per step, it may be necessary to use the results of Theorem 3.2, Corollary 4.1, and § 4.4. Subscripts and hats are omitted from the α columns for convenience; temporary vectors $s = Cr$ and $q = C^{-1}p$ are used for brevity.

We now discuss each of the CG patterns:

GCGHS (*Generalized CGHS*). This pattern generalizes the classical conjugate gradient method of Hestenes and Stiefel, which we call CGHS. It is distinguished by its use of the preconditioned system matrix for the inner product matrix, that is, $B = CA$. Since B must be hpd, this severely limits the choices for C.

GCR (*Generalized Conjugate Residuals*). This pattern generalizes the conjugate residual method. Here $B = (CA)^*(CA)$, which is hpd for any nonsingular CA. Although CA is B-self-adjoint if and only if CA is Hermitian, this method has important applications when CA is B-normal(1), as will be shown in § 5.4. For this reason we

TABLE 5.1
Seven CG *patterns.*

Pattern	Name	B	CA	Orthodir		Orthomin	
				Restrictions	α	Restrictions	$\hat{\alpha}$
P1	GCGHS	CA	CA	CA hpd	$\frac{\langle s,p\rangle}{\langle CAp,p\rangle}$	CA hpd	$\frac{\langle s,s\rangle}{\langle CAp,p\rangle}$
P2	GCR	$(CA)^*(CA)$	CA	CA herm	$\frac{\langle s,CAp\rangle}{\langle CAp,CAp\rangle}$	CA hpd	$\frac{\langle s,CAs\rangle}{\langle CAp,CAp\rangle}$
P3	GPCG	EA	DEA	EA hpd D herm	$\frac{\langle Er,p\rangle}{\langle EAp,p\rangle}$	EA hpd D hpd	$\frac{\langle Er,s\rangle}{\langle EAp,p\rangle}$
P4	GPCR	A^*EA	CA	E hpd EAC herm	$\frac{\langle Er,Ap\rangle}{\langle EAp,Ap\rangle}$	E hpd EAC hpd	$\frac{\langle Er,As\rangle}{\langle EAp,Ap\rangle}$
P5	GCGE	I	A^*DA	D herm	$\frac{\langle r,Dq\rangle}{\langle p,p\rangle}$	D hpd	$\frac{\langle r,Dr\rangle}{\langle p,p\rangle}$
P6	GCGIB	B	$B^{-1}A^*DA$	B hpd D herm	$\frac{\langle r,Dq\rangle}{\langle Ap,Dq\rangle}$	B hpd D hpd	$\frac{\langle r,Dr\rangle}{\langle DAp,q\rangle}$
P7	GCGCB	B	A^*DA	B hpd BD herm $AB=BA$	$\frac{\langle Br,Dq\rangle}{\langle BAp,Dq\rangle}$	B hpd BD hpd $AB=BA$	$\frac{\langle Br,Dq\rangle}{\langle BAp,Dq\rangle}$

use the notation $(CA)^*$. Notice that this pattern is applicable to a larger class of matrices than GCGHS, but minimizes in a different norm.

GPCG (*Generalized Preconditioned CG*). This pattern is a generalization of the preconditioned conjugate gradient method. Here we assume $B = EA$ and $C = DE$, where D and E are computable matrices. By choosing $C = DE$ it is easy to see that BCA is Hermitian whenever D is Hermitian. Note that GCGHS is a special case of GPCG.

GPCR (*Generalized Preconditioned CR*). This pattern generalizes the preconditioned conjugate residual method by using $B = A^*EA$, where E is a computable hpd matrix. When $E = C$, the method PCR results. This pattern is also important for the commutative preconditioned methods considered in § 5.3. The notation A^* is again used in anticipation of applications for non-Hermitian A; see § 5.4. Note that GCR is a special case of GPCR.

GCGE (*Generalized CG with Error minimization*). This method is a generalization of Craig's method for the normal equations (§ 5.2). Unlike the previous patterns, this pattern does not assume $B = ZA$ for some computable Z. Instead, the error is minimized in the $B = I$ norm. To obtain a computable method, however, we must assume $C = A^*D$ for some computable matrix D. By using a commutative preconditioner it is possible to minimize in the I-norm without resorting to some variant of the normal equations; see § 5.3. Also observe that this pattern is the most general possible if we wish to use $B = I$.

GCGIB (*Generalized CG with Invertible B*). This pattern assumes neither $B = ZA$ nor $B = I$. Nevertheless, α_i is computable because $\langle Be_i, Cq_i\rangle = \langle r_i, Dq_i\rangle$ is computable; recall equation (4.5). A similar result holds for $\hat{\alpha}_i$. Since C must be computable, B^{-1} must be computable, and so the choices for B are limited. Notice, however, that B need *not* be computable.

GCGCB (*Generalized CG with Commutative B*). This pattern is similar to GCGIB,

TABLE 5.2
Ten CG *methods.*

Method (Pn)	B	CA	Odir Restrictions	Omin Restrictions
CGHS (P1)	A	A	A hpd	A hpd
CR (P2)	A^2	A	A herm	A hpd
PCG (P3)	A	CA	A hpd, C herm	A hpd, C hpd
PCR (P4)	ACA	CA	A herm, C hpd	A hpd, C hpd
CGNR (P1)	A^*A	A^*A	none	none
CGNE (P5)	I	A^*A	none	none
PCGNS (P1)	$(M^{-1}A)^*M^{-1}A$	$(M^{-1}A)^*M^{-1}A$	none	none
PCGNE (P5)	I	$(M^{-1}A)^*M^{-1}A$	none	none
PCGNR (P3)	A^*A	$(M^*M)^{-1}A^*A$	none	none
PCGNM (P6)	M^*M	$(M^*M)^{-1}A^*A$	none	none

but does not use B^{-1} in C. Instead, we assume $BA = AB$. This guarantees that the pattern is computable because $C^*Be_i = D^*ABe_i = D^*BAe_i = D^*Br_i$ is computable. It is also easy to show that CA is B-self-adjoint if and only if D is B-self-adjoint, that is, if and only if BD is Hermitian. This is most readily achieved when B and D are real polynomials in the matrix A; see § 5.3. This pattern has not appeared in the literature, except for the special case $B = I$, in which case GCGCB and GCGIB both reduce to GCGE.

In this section we have identified seven basic CG patterns. By specifying the matrices in these patterns, particular methods result. Although the patterns do not encompass every possible CG method, they do suggest how new methods might be designed. Moreover, these patterns make it possible to see more clearly the relationship between various CG methods. In the next subsection we show how ten known CG methods fit into the above taxonomy. In particular, we show that each method is an instance of a pattern. Of course, since the patterns in Table 5.1 are not mutually exclusive, some methods are instances of more than one pattern.

5.2. Ten CG methods. In this section we show how ten well-known CG methods fit into our taxonomy. These methods are summarized in Table 5.2, where we list the inner product matrix B and preconditioned system matrix CA. We also list sufficient conditions for the convergence of the Odir and Omin algorithms. In particular, for Odir, we give sufficient conditions for B to be hpd and for BCA to be Hermitian. For Omin, we give sufficient conditions for both B and BCA to be hpd. The primary pattern of which the method is an instance is given in parentheses; refer to Table 5.1.

CGHS. The best-known CG method is the original conjugate gradient method of Hestenes and Stiefel [21], which we call CGHS. Here $C = I$ and $B = A$, so at each step $\|e_i\|_A$ is minimized over $x_0 + V_i(r_0, A)$. For $B = A$ to define an inner product norm, A must be hpd, a well-known requirement. Since $BCA = A^2$ is hpd, Omin may be used, which gives the classical CG algorithm. This method is an instance of GCGHS ($C = I$).

CR. The *conjugate residual* method [21], which we call CR, may be used for both Hermitian positive definite and indefinite matrices A. This is so because $B = A^2$ is hpd for any nonsingular Hermitian A. Since $\langle Be_i, e_i\rangle = \langle r_i, r_i\rangle$, the I-norm of the

residual is minimized at each step, and for this reason CR is often called the *minimum residual* method. Note that $BCA = A^3$ is hpd only if A is hpd, and so Odir should be used for indefinite A. This method is an instance of pattern GCR ($C = I$).

PCG. Closely related to CGHS is the well-known *preconditioned CG* method, which we call PCG. This method, described in [10] and [28], also minimizes in the $B = A$ norm, but has preconditioned system matrix CA. The matrix C is often expressed as $C = M^{-1}$, where M^{-1} is an approximation to A^{-1}. Although M commonly represents an incomplete LU factorization of A, it may be any matrix splitting, for example, Jacobi, Gauss–Seidel, SSOR, or ADI. In its usual Omin formulation, PCG requires an hpd M, and is then equivalent to CGHS on $\tilde{A} = M^{-1/2}AM^{-1/2}$. However, if Odir is used, M may be Hermitian indefinite. This method is an instance of GPCG ($E = I$, $D = C$).

PCR. The fourth method is *preconditioned conjugate residuals*, or PCR [6]. Here $B = ACA$, and so the preconditioner C must be hpd, even though A may be indefinite. In contrast, PCG requires an hpd A but allows C to be indefinite. As with CR, Omin should be used only if A is hpd. This method is an instance of GPCR ($E = C$).

Normal equations. When A is not B-normal(1), Odir or Omin may still be employed, but only if applied to the normal equations.[2] Of course, a preconditioning also may be used. By arranging the normal equations in different ways, different algorithms result; six are listed in Table 5.2 and described below. See also [15].

CGNR. This method, which is an instance of pattern GCGHS ($C = A^*$), is equivalent to CGHS on $A^*Ax = A^*b$. Consequently, it minimizes the I-norm of the residual at each step. It differs from CR in that its Krylov space is generated by A^*A rather than A. Note that CGNR is also an instance of patterns P3 and P4.

CGNE. This method, which is often called Craig's method [11], is equivalent to CGHS on $AA^*y = b$. It differs from CGNR in that it minimizes the I-norm of the *error* at each step. Again the Krylov space is generated by A^*A. This method is an instance of pattern GCGE ($D = I$).

PCGNS and PCGNE. These methods result from applying CGNR and CGNE to $M^{-1}A$, where M^{-1} is an *inner* preconditioning. The total preconditioning is $C = (M^{-1}A)^*M^{-1}$. Note that PCGNS minimizes the I-norm of the preconditioned residual whereas PCGNE minimizes the I-norm of the error. We consider PCGNS to be primarily an instance of pattern GCGHS ($C = (M^{-1}A)^*M^{-1}$), but it is also an instance of P3 and P4. PCGNE is an instance GCGE ($D = (MM^*)^{-1}$).

PCGNR and PCGNM. These methods are equivalent to CGNR and CGNE on AM^{-1}. Here the total preconditioning is $C = (M^*M)^{-1}A^*$. This choice for C is motivated by the assumption that if M^{-1} is a good preconditioning for A, then $(M^*M)^{-1}$ is a good preconditioning for A^*A. Since PCGNR also results from PCG by replacing C with $(M^*M)^{-1}$ and A with A^*A, we consider it to be an instance of pattern GPCG ($E = A^*$, $D = (M^*M)^{-1}$). However, it is also an instance of P4. PCGNM is an instance of GCGIB ($B = M^*M$, $D = I$).

In this section we have examined ten known CG methods and shown how each fits into the taxonomy of § 5.1. In particular, we have shown that each is an instance of one or more CG pattern. Once the pattern was recognized, it was easy to obtain sufficient conditions for the convergence of Odir and Omin from Table 5.1. We have

[2] This taxonomy considers only 3-term CG methods. Truncated and restarted CG methods, as well as Orthogonal Error Methods, are briefly discussed in § 5.5.

illustrated each of the patterns of § 5.1 except GCGCB. We will furnish an example of it in the next section where we discuss the role of commutativity in CG methods.

5.3. Commutativity and CG methods. When certain commutativity properties among the matrices B, C, and A hold, it is often easier to satisfy the sufficient conditions listed in Table 5.1. For example, suppose C and A commute, a case we call *commutative preconditioning*. Then CA is Hermitian (hpd) whenever C and A are Hermitian (hpd). This makes practicable several CG methods based on GCGHS, GCR, and GPCR. Methods based on other commutativity properties are also possible; see Table 5.3 below.

Before discussing these methods, let us consider how a commutative preconditioner C might be found for a B-normal matrix A. First observe that if C is also B-normal and commutes with A, then CA is B-normal, and so an optimal CG method exists for CA. For the method to be economical, however, CA must have normal degree at most 1. This requirement limits the choice of C because the normal degree of CA depends on C. Nevertheless, there are at least two important special cases:

(1) Suppose $C = C(\lambda)$ is an analytic function in a region containing the spectrum of A. Then $C(A)$ is B-normal and commutes with A. Moreover, if C is chosen so that for each λ_i in the spectrum of A, $C(\lambda_i)\lambda_i$ lies on a straight line in the complex plane, then $C(A)A$ is B-normal(1). Although finding such a C is difficult in general, it is possible: If A is B-self-adjoint and $C(\lambda)$ is real valued on the real line, then $C(A)A$ is B-self-adjoint (and so has real eigenvalues). For example, let C be a real polynomial in λ. This is *polynomial preconditioning*, for which several CG methods are possible (see below). The importance of polynomial preconditioning for vector machines has been demonstrated in [2], [4], [13], [24], [25]. On some parallel machines, where inner products are more costly than matrix-vector multiplications, the potential advantages are even greater [5], [29].

(2) Let A be Hermitian and consider the ADI-like splitting, $A = H + V$. It yields the preconditioning matrix

$$C = (H + \eta I)^{-1}(V + \nu I)^{-1}$$

where η and ν are ADI parameters; see, e.g., [8]. If H and V commute, then C is Hermitian and commutes with A. Therefore, CA is Hermitian. If H and V are also hpd, and $\eta, \nu \geq 0$, then CA is hpd.

Methods for commuting C and A. Suppose C and A are Hermitian and commute. Then CA is Hermitian and pattern GCR is applicable; it minimizes the I-norm of the preconditioned residual. The method PCRI, which is an instance of pattern GPCR ($E = I$), may also be used. This method is similar to GCR, but minimizes the I-norm of the residual at each step.

Recall that GCGHS is applicable whenever CA is hpd. Since C and A commute, this is satisfied whenever C and A are hpd. Suppose, however, that A is Hermitian indefinite. Then GCGHS is applicable if and only if C is such that CA is hpd. Finding such a C is possible by choosing C to be, for example, a polynomial in A. If C is a good preconditioning for A, CA is an approximation to the identity matrix, and so GCGHS approximately minimizes the I-norm of the error at each step.

Other commutativity properties. So far we have considered only commutative preconditioning. We now discuss two related commutativity properties and some methods that depend on them. These methods are listed in Table 5.3. As in Table 5.2,

TABLE 5.3
Commutativity in CG *methods.*

Method (Pn)	B	CA	Odir Restrictions	Omin Restrictions
PPCR (P4)	$A^*\Gamma^2 A$	$A^*\Gamma A$	Γ herm, $\Gamma A = A\Gamma$	Γ hpd, $\Gamma A = A\Gamma$
CGCB (P7)	Γ	$A^*\Gamma A$	Γ hpd, $\Gamma A = A\Gamma$	Γ hpd, $\Gamma A = A\Gamma$
PPCGM (P4)	$A^*M^{-1}A$	$S(M^{-1}A)M^{-1}A$	M hpd $AS(M^{-1}A)$ herm	M hpd $AS(M^{-1}A)$ hpd
PPCGSM (P3)	$S(AM^{-1})A$	$S(M^{-1}A)M^{-1}A$	M herm $S(AM^{-1})A$ hpd	M hpd $S(AM^{-1})A$ hpd
PPCGIB (P6)	M	$M^{-1}A\Gamma(M^{-1}A)M^{-1}A$	M hpd, A herm	M hpd, A herm $\Gamma(M^{-1}A)M^{-1}$ hpd

sufficient conditions are given for the convergence of the Odir and Omin algorithms. If these conditions are satisfied, then so are the sufficient conditions listed in Table 5.1.

In the first two methods we assume $C = A^*\Gamma$, where Γ is Hermitian and commutes with A. These assumptions may seem unusual, but they are easily satisfied when A is Hermitian and C is a polynomial in A. Here $C(\lambda)\lambda = \Gamma(\lambda)\lambda^2$, and so $C(\lambda)\lambda$ has a double root at the origin. This leads to several interesting CG methods [2]. More generally, the assumptions are satisfied when A is I-normal (with possibly complex eigenvalues) and $\Gamma(\lambda)$ is real-valued on the spectrum of A. Although our emphasis is on Hermitian A, A^* is used for greater generality. Note that C and A commute when A is I-normal, that is, when $A^*A = AA^*$.

PPCR. This method is an instance of GPCR ($E = \Gamma^2$, $C = A^*\Gamma$). Note that EAC is Hermitian whenever $\Gamma A = A\Gamma$. If A is Hermitian, then so is C, in which case the method is biased toward the small eigenvalues of A^2.

CGCB. This new method, which is an instance of GCGCB ($B{=}D{=}\Gamma$), exploits the commutativity of A and Γ to obtain computability. Specifically, C^*Be_i is computable because $C^*Be_i = \Gamma A\Gamma e_i = \Gamma^2 Ae_i$ is computable. If C is a good approximation to A, then $CA = A^*\Gamma A$ is an approximation to I, and so Γ is an approximation to $(AA^*)^{-1}$. Thus, this method is biased toward the small *singular* values of A. Note that Omin is always applicable because Γ must be hpd to define a norm.

In addition to PPCR and CGCB, two other methods are practicable when $C = A^*\Gamma$. Although these methods do not require the commutativity of A and Γ, we mention them here to emphasize their use in polynomial preconditioning for Hermitian A. The first method is GCGE with $D = \Gamma(A)$, which is Hermitian because A is Hermitian. This method allows us to minimize the Euclidean norm of the error without resorting to some form of the normal equations (recall CGNE). The second method is PCGA, which results from pattern GPCG ($E = \Gamma$, $D = A$). Here Γ must be chosen so that ΓA is hpd. Since $B = C$ is an approximation to A^{-1}, PCGA gives greater weight to the small eigenvalues of A. PCG, on the other hand, is biased toward the large eigenvalues of A.

In the last three methods of Table 5.3, A is first preconditioned by some M^{-1}, and then $\hat{A} = M^{-1}A$ is preconditioned by a polynomial in $\hat{A}$. The total preconditioning for A is

$$C = S(M^{-1}A)M^{-1} = M^{-1}S(AM^{-1})$$

where $S(\lambda)$ is the polynomial preconditioner for $\hat{A}$. Clearly, $S(\hat{A})$ and $\hat{A}$ commute, which is the commutativity property employed here. Note that C and A commute when $M^{-1}A = AM^{-1}$. Also note that C is Hermitian whenever A and M are Hermitian. Thus, if A is hpd, the method PCG may be used, an observation first made in [1]. If C is chosen so that CA is Hermitian (hpd), then GCR (GCGHS) is applicable. Several other methods are also possible, three of which are given in Table 5.3.

PPCGM. This method is an instance of GPCR ($E = M^{-1}$, $C = S(M^{-1}A)M^{-1}$), and it is applicable whenever M is hpd and $AS(M^{-1}A)$ is Hermitian. The latter is true whenever A is Hermitian.

PPCGSM. This method is an instance of GPCG ($D = M^{-1}$, $E = S(AM^{-1})$). Since $B = S(AM^{-1})A = AS(M^{-1}A)$ must be hpd, the polynomial S must be chosen more carefully than in PPCGM. On the other hand, M only needs to be Hermitian.

PPCGIB. Analogous to our earlier assumption, suppose $S(\lambda) = \Gamma(\lambda)\lambda$. Then a method based on GCGIB is possible. In particular, let $B = M$ and $D = \Gamma(M^{-1}A)M^{-1}$. For computability we require that A be Hermitian, which implies that D is Hermitian. Note that M need not be computable.

5.4. CG methods for nonself-adjoint systems. In the tables above, sufficient conditions were given for the preconditioned system matrix CA to be B-self-adjoint. A variety of methods, that is, instances of patterns, were examined. In this section we explore the *full* domain of applicability for each pattern listed in Table 5.1. To do this we give necessary and sufficient conditions for CA to be B-normal(1), the most general class of matrices for which a 3-term CG method exists. Of course, these conditions also apply to the methods based on these patterns.

The domain of applicability for pattern P1 (GCGHS) has already been fully explored; the requirement that CA be hpd is inviolable. The domains of applicability for patterns P3, P5, P6, and P7 are slightly larger than indicated in Table 5.1, but not significantly so. Only in patterns P2 (GCR) and P4 (GPCR) are the domains much larger than given in Table 5.1. Moreover, the method PCR (an instance of GPCR) is the only method presented in this paper that is applicable to *any* definite matrix without resorting to some form of the normal equations.

Before presenting Table 5.4 and discussing each pattern, we first prove the following theorem.

THEOREM 5.1. *Let B be hpd. Then CA is B-normal(1) if and only if*

(i) $e^{i\theta}BCA$ *is Hermitian for some* θ; *or*
(ii) $2B = \gamma BCA + (\gamma BCA)^*$ *for some (possibly complex)* γ.

Proof. Recall that CA is B-normal(1) if and only if $CA = e^{i\theta}(i\frac{r}{2}I + G)$ for some $r \geq 0$, $0 \leq \theta \leq 2\pi$, and $G^+ = G$. This yields $(CA)^+ = e^{-2i\theta}CA - ire^{-i\theta}I$. Thus, CA is B-normal(1) if and only if $irI = e^{-i\theta}CA - (e^{-i\theta}CA)^+$. Suppose first that $r = 0$. Then $(e^{-i\theta}CA)^+ = e^{-i\theta}CA$. In other words, CA is only a rotation away from being B-self-adjoint. Note that this is true if and only if $(e^{-i\theta}BCA)^* = e^{-i\theta}BCA$. Next suppose that $r \neq 0$. Then $2I = \gamma CA + (\gamma CA)^+$ where $\gamma = -i\frac{2}{r}e^{-i\theta}$. This is true if and only if $2B = \gamma BCA + (\gamma BCA)^*$. □

In Table 5.4 we list for each CG pattern necessary and sufficient conditions for the use of Odir. These conditions follow from Theorem 5.1 and the requirement that B be hpd. Necessary and sufficient conditions for use of the cheaper Omin algorithm are also given. We denote the Hermitian part of G by $\mathcal{H}(G) = \frac{1}{2}(G + G^*)$ in the table and text below.

TABLE 5.4
Domains of applicability for the seven CG *patterns;* $\mathcal{H}(G) = \frac{1}{2}(G + G^*)$.

Pn	Name	B	CA	Odir Restrictions	Omin Restrictions
P1	GCGHS	CA	CA	CA hpd	CA hpd
P2	GCR	$(CA)^*(CA)$	CA	CA I-normal(1)	CA I-normal(1) CA definite
P3	GPCG	EA	DEA	EA hpd $e^{i\theta}D$ herm *or* $(EA)^{-1} = \mathcal{H}(\gamma D)$	Odir restrictions *and* D definite
P4	GPCR	A^*EA	CA	E hpd $e^{i\theta}EAC$ herm *or* $2E = \mathcal{H}(\gamma EAC)$	Odir restrictions *and* EAC definite
	PCR	A^*CA	CA	C hpd $e^{i\theta}A$ herm *or* $C^{-1} = \mathcal{H}(\gamma A)$	Odir restrictions *and* A definite
P5	GCGE	I	A^*DA	$e^{i\theta}D$ herm *or* $I = A^*\mathcal{H}(\gamma D)A$	Odir restrictions *and* D definite
P6	GCGIB	B	$B^{-1}A^*DA$	B hpd $e^{i\theta}D$ herm *or* $B = A^*\mathcal{H}(\gamma D)A$	Odir restrictions *and* D definite
P7	GCGCB	B	A^*DA	B hpd $AB = BA$ $e^{i\theta}BD$ herm *or* $B = A^*\mathcal{H}(\gamma BD)A$	Odir restrictions *and* BD definite

GCGHS. Since $B = CA$ must be hpd to define a norm, CA is necessarily B-normal(1). In fact, CA is B-self-adjoint. Thus the domain of applicability for GCGHS is the set of all CA such that CA is hpd.

GCR. This method is applicable if and only if CA is $(CA)^*(CA)$-normal(1), which is true if and only if CA is I-normal(1). In other words, $CA = e^{i\theta}(i\frac{r}{2}I + G)$ for some $r \geq 0$, $0 \leq \theta \leq 2\pi$, and *Hermitian* G. For example, suppose $C = I$. Matrices of the form $A = \mu I + R$, where μ is complex and R is real Hermitian (i.e., real symmetric), arise in electromagnetics. Matrices of the form $A = \mu I + S$, μ real and S skew-Hermitian ($S^* = -S$), arise in quantum chromodynamics.

GPCG. As in Table 5.1, $B = EA$ must be hpd. For B-normality(1) we require that either $e^{i\theta}D$ is Hermitian for some θ or that $(EA)^{-1} = \mathcal{H}(\gamma D)$ for some γ. The first condition requires that D be essentially Hermitian; the second condition is obscure and appears difficult to satisfy. Thus, this pattern benefits little from considering B-normal(1) instead of B-self-adjoint matrices.

We remark that the GCW method of Concus and Golub [9] and Widlund [34] is an instance of this pattern with $E = I$ and $C^{-1} = \mathcal{H}(A)$. However, this choice of E does *not* yield an hpd B unless A is itself hpd. Therefore, GCW is not a CG method, but instead an Orthogonal Error Method; see also [14] and [17]. However, if A is real and we consider the odd iterates of GCW, the resulting method is an instance of GCGIB [19]. See below.

GPCR. As in Table 5.1, E must be hpd. This pattern is applicable if and only if CA is A^*EA-normal(1), which is true if and only if AC is E-normal(1). Thus, by Theorem 5.1, we require that either $e^{i\theta}EAC$ is Hermitian for some θ or that $E = \mathcal{H}(\gamma EAC)$ for some γ. The first condition is essentially the same as given in Table 5.1. The second condition is, in general, too obscure to be useful. Two important special cases are

$E = I$ and $E = C$. In the former case AC must be I-normal(1). If C is Hermitian and commutes with A, this is true whenever A is I-normal(1). For example, if C is a polynomial in the Hermitian matrix A, $AC = CA$ is Hermitian, and so both GPCR and GCR are applicable. Also note that GPCR and GCR are identical when $C = I$.

The Method PCR. When $E = C$ in GPCR, the important method PCR results; see Table 5.4. Here we require that C be hpd so that $B = A^*CA$ is hpd. This method is applicable if and only if AC is C-normal(1). By Theorem 5.1, this is equivalent to requiring that either $e^{i\theta}A$ be Hermitian for some θ or that $C^{-1} = \mathcal{H}(\gamma A)^*$ for some γ. The first condition is essentially the same as given in Table 5.2. The second condition is more interesting: it implies that PCR is applicable to any definite matrix.

To further explain this choice of C, let $S = \frac{1}{2}(A + A^*)$ and $N = \frac{1}{2}(A - A^*)$; also let $\gamma = \xi + i\eta$. Then $C = (\xi S + i\eta N)^{-1}$. In other words, C is the inverse of some linear combination of S and iN, both of which are Hermitian. Since C must be hpd, not every linear combination suffices. A more enlightening view of C is to note that it must be the inverse of the Hermitian part of γA for some $\gamma \neq 0$. If A is definite, γ may be chosen so that $S_\gamma = \frac{1}{2}(\gamma A + \bar{\gamma} A^*)$ is hpd. If $N_\gamma = \frac{1}{2}(\gamma A - \bar{\gamma} A^*)$, then

$$B = A^*CA = A^*S_\gamma^{-1}A = \frac{1}{|\gamma|^2}(S_\gamma + N_\gamma^* S_\gamma^{-1} N_\gamma)$$

is also hpd. Moreover,

$$CA = \frac{1}{\gamma}(I + S_\gamma^{-1}N_\gamma).$$

The use of this method with $\gamma = 1$ was first described in [15].

GCGE. In this pattern $B = I$, and so the I-norm of the error is minimized at each step. Theorem 5.1 requires that either $e^{i\theta}D$ be Hermitian or that $\mathcal{H}(\gamma D) = (AA^*)^{-1}$. Although the second condition is impractical, the first is interesting. It states that D must be essentially Hermitian, the condition given in Table 5.1. Note that there is no restriction on the matrix A. If D is hpd, then $A^*DA = (D^{1/2}A)^*D^{1/2}A$ is a form of the normal equations. For example, if $D = I$, the method CGNE results. Another choice is $D = (MM^*)^{-1}$, where M^{-1} is any preconditioning for A. This yields the method PCGNE of Table 5.2. If D is Hermitian indefinite, we may minimize in the $B = I$ norm without resorting to some form of the normal equations; recall § 5.3.

GCGIB. Here B is an arbitrary hpd matrix. The restrictions on D are similar to those in GCGE. Again note that A may be arbitrary if $e^{i\theta}D$ is Hermitian for some θ. For example, suppose $B = S_\gamma = \mathcal{H}(\gamma A)$ for some γ for which S_γ is hpd. Furthermore, let $D = S_\beta^{-1}$ for any $\beta \neq 0$. Then

$$CA = B^{-1}A^*DA = \frac{1}{\bar{\gamma}\beta}S_\gamma^{-1}(S_\gamma - N_\gamma)S_\beta^{-1}(S_\beta + N_\beta) = \frac{1}{\bar{\gamma}\beta}(I - S_\gamma^{-1}N_\gamma)(I + S_\beta^{-1}N_\beta).$$

If $\beta = \gamma$, then

$$CA = \frac{1}{|\gamma|^2}\left(I - (S_\gamma^{-1}N_\gamma)^2\right).$$

Compare this method to PCR with $C = S_\gamma^{-1}$; there $|\gamma|^2 B = S_\gamma + N_\gamma^* S_\gamma^{-1} N_\gamma$ and $\gamma CA = I + S_\gamma^{-1}N_\gamma$. Note the similarity between the inner product matrices. Although GCGIB requires twice as many matrix-vector multiplications as PCR, there may be circumstances for which each method is preferable.

If A is real and definite, S_γ is hpd for either $\gamma = 1$ or $\gamma = -1$. Hageman, Luk, and Young [19] have shown that GCGIB with $B = S_\gamma$ and $D = S_\gamma^{-1}$ is equivalent to the "double" method obtained by taking every other step of the GCW method. Also see GPCG above.

GCGCB. Again B must be hpd; we require that it commute with A for computability. (If $B = ZA$ for some computable matrix Z, see patterns P1–P4.) The restrictions on BD follow from Theorem 5.1. The first condition is essentially the same as given in Table 5.1; the second condition seems intractable. Also see § 5.3.

In this section we have explored the domain of applicability for each of the patterns introduced in Table 5.1. For the most part, these patterns are limited to B-self-adjoint matrices. However, GCR and PCR have broader application. For example, PCR is applicable to *any* definite matrix A. Of course, the efficacy of this method depends on the quality of $C = (\mathcal{H}(\gamma A))^{-1}$ as a preconditioner. If this C is a poor preconditioner, some form of the normal equations might be more efficient.

5.5. Other methods. The methods above assume CA is B-normal(1). If CA is not B-normal(1) for some B, a 3-term CG method *does not exist.* For such matrices it is tempting to use CG(B, C, A) anyway. The resulting method is called a *truncated* CG method, so named because the recursion (2.3) is artificially truncated. Although not optimal at each step, such methods are often effective [15], [35]. Along with optimality, the equivalence of Odir, Omin, and Ores is lost. In practice, truncated Omin and Ores have been observed to converge faster than truncated Odir [35]. A CG method may also be *restarted* after s steps. This leads to a cyclic method; within a cycle the error is minimized at each step. An example of such a method is GMRES [33]. See [32] for a discussion and comparison of truncated and/or restarted methods.

Finally, we might consider an Orthogonal Error Method [17]. Here the matrix B need only be definite. Like CG methods, an OEM is guaranteed to converge in at most n steps, and a 3-term OEM exists for CA if and only if CA is B-normal(1) or $d(CA) \leq 3$. Since B is no longer hpd, the definition of B-normal(1) is slightly different; cf. [17]. Moreover, it is unclear whether an OEM reduces the error in *some* norm at every step, as does a CG method.

6. Eigenvalue estimates. It is well known [18] that CGHS is related to the Lanczos procedure for computing eigenvalues. A similar relationship exists for PCG [10]. In this section we generalize the relationship to any CG method: When CA is B-normal(1), eigenvalue estimates for CA may be obtained from a Hermitian tridiagonal matrix constructed from the iteration parameters γ_j and σ_j (if Odir is used), or $\hat{\alpha}_j$ and β_j (if Omin is used). Moreover, these estimates lie in $H(CA)$, the convex hull of the spectrum of CA. Since CA is B-normal(1), $H(CA)$ is a line segment in the complex plane. If this line segment does not include the origin, then estimating the B-condition number of CA,

$$\kappa_B(CA) = \|CA\|_B \|(CA)^{-1}\|_B,$$

is equivalent to determining the endpoints of $H(CA)$. We will use this estimate of $\kappa_B(CA)$ in the stopping criteria discussed below.

6.1. Preliminaries. Suppose CA is B-normal(1) and $p_0, \ldots, p_{k-1}$ are the B-orthogonal direction vectors generated by Odir. Also let P_k be the $n \times k$ matrix with columns $p_0, \ldots, p_{k-1}$. Theorem 3.2b implies

$$P_k^* B P_k = D_k = \operatorname{diag}(\delta_{j-1}), \quad \delta_{j-1} = \langle Bp_{j-1}, p_{j-1} \rangle.$$

Since B is hpd, δ_{j-1} is positive, and so D_k is hpd. The three term recursion (3.2g) yields

$$P_k^* BCAP_k = D_k T_k$$

where $T_k = \text{tridiag}\,(1, \gamma_{j-1}, \sigma_j)$ is a tridiagonal matrix.

The *field of values* of G with respect to the B inner product is defined to be

$$F_B(G) = \left\{ \lambda : \lambda = \frac{\langle BGx, x\rangle}{\langle Bx, x\rangle} \text{ for some } x \in \boldsymbol{C}^n \right\}.$$

Now suppose λ is an eigenvalue of T_k with eigenvector x, that is, $T_k x = \lambda x$. Then

$$\lambda = \frac{\langle D_k T_k x, x\rangle}{\langle D_k x, x\rangle} = \frac{\langle BCAP_k x, P_k x\rangle}{\langle BP_k x, P_k x\rangle} \in F_B(CA).$$

This leads to the main result of this section.

THEOREM 6.1. *If λ is an eigenvalue of T_k, then $\lambda \in H(CA)$.*

Proof. The result follows from the discussion above and the fact that $F_B(CA) = H(CA)$ if CA is B-normal; cf. [22]. □

Theorem 6.1 guarantees that the eigenvalues of T_k lie in $H(CA)$. However, using a remarkable property of tridiagonal matrices, much more can be said. To do so it is necessary to examine T_k in greater detail.

6.2. The matrix T_k. Recall from equation (2.8) that CA is B-normal(1) if and only if

$$CA = e^{i\theta}\left(i\frac{r}{2}I + G\right), \quad r \geq 0, \quad 0 \leq \theta \leq 2\pi, \quad G^+ = G. \tag{6.1}$$

Therefore,

$$D_k T_k = P_k^* BCAP_k = e^{i\theta}\left(i\frac{r}{2}D_k + P_k^* BGP_k\right).$$

Now define

$$\tilde{T}_k = D_k^{1/2} T_k D_k^{-1/2} = D_k^{-1/2} P_k^* BCAP_k D_k^{-1/2} = e^{i\theta}\left(i\frac{r}{2}I + H_k\right) \tag{6.2}$$

where $H_k = D_k^{-1/2} P_k^* BGP_k D_k^{-1/2}$. Since G is B-self-adjoint, $BG = G^* B$, and so H_k is Hermitian. Thus $\tilde{T}_k$ is the translation and rotation of a Hermitian matrix.[3] Moreover, since T_k and $\tilde{T}_k$ are similar, they have the same eigenvalues. To find the eigenvalues of T_k it is convenient to find those of H_k first. Once these are computed, the eigenvalues of T_k are easily found by translation and rotation. Also notice that (6.2) resembles (6.1): H_k is Hermitian in (6.2) whereas G is B-self-adjoint in (6.1).

We now show how the Odir iteration parameters, γ_j and σ_j, may be used to construct the matrix H_k. Recall that

$$T_k = \text{tridiag}\,(1, \gamma_{j-1}, \sigma_j)$$

[3] It can also be shown that $\tilde{T}_k$ is an orthogonal section of $B^{1/2}CAB^{-1/2}$, which is similar to CA. This provides an alternative means of obtaining the results in this section.

which yields

$$\tilde{T}_k = \text{tridiag}\left(\left(\frac{\delta_{j-1}}{\delta_{j-2}}\right)^{1/2},\ \gamma_{j-1},\ \sigma_j\left(\frac{\delta_{j-1}}{\delta_j}\right)^{1/2}\right). \tag{6.3}$$

From Corollary 3.3e and its proof it follows that

$$\sigma_j\left(\frac{\delta_{j-1}}{\delta_j}\right)^{1/2} = e^{2i\theta}\left(\frac{\delta_j}{\delta_{j-1}}\right)^{1/2} \quad \text{and} \quad \frac{r}{2} = \text{Im}(e^{-i\theta}\gamma_{j-1}).$$

Substituting these into expression (6.3) for $\tilde{T}_k$ gives

$$H_k = \text{tridiag}\left(e^{-i\theta}\left(\frac{\delta_{j-1}}{\delta_{j-2}}\right)^{1/2},\ \text{Re}\,(e^{-i\theta}\gamma_{j-1}),\ e^{i\theta}\left(\frac{\delta_j}{\delta_{j-1}}\right)^{1/2}\right). \tag{6.4}$$

When CA is B-self-adjoint, $r = 0$ and $\theta = 0$. Corollary 3.3e then yields $\sigma_j = \delta_j/\delta_{j-1}$, and equation (6.2) becomes

$$\tilde{T}_k = H_k = \text{tridiag}\,(\sqrt{\sigma_{j-1}},\ \gamma_{j-1},\ \sqrt{\sigma_j}). \tag{6.5}$$

Here $\sigma_j > 0$, so the square roots are real, and thus $\tilde{T}_k$ is real symmetric.

In the derivation above, H_k is defined in terms of γ_j and σ_j, the iteration parameters of Odir. When BCA is definite, Omin should be employed. Equation (4.2) may be used to redefine H_k in terms of $\hat{\alpha}_j$ and β_j, the iteration parameters of Omin:

$$T_k = \text{tridiag}\left(-\frac{1}{\hat{\alpha}_{j-2}},\ \frac{1+\beta_{j-1}}{\hat{\alpha}_{j-1}},\ -\frac{\beta_{j-1}}{\hat{\alpha}_j}\right),$$

$$\tilde{T}_k = \text{tridiag}\left(-\left(\frac{\delta_{j-1}}{\delta_{j-2}}\right)^{1/2}\frac{1}{\hat{\alpha}_{j-2}},\ \frac{1+\beta_{j-1}}{\hat{\alpha}_{j-1}},\ -\left(\frac{\delta_{j-1}}{\delta_j}\right)^{1/2}\frac{\beta_{j-1}}{\hat{\alpha}_j}\right),$$

and

$$H_k = \text{tridiag}\left(-e^{-i\theta}\left(\frac{\delta_{j-1}}{\delta_{j-2}}\right)^{1/2}\frac{1}{\hat{\alpha}_{j-2}},\ \text{Re}\left(e^{-i\theta}\,\frac{1+\beta_{j-1}}{\hat{\alpha}_{j-1}}\right),\ -e^{i\theta}\left(\frac{\delta_j}{\delta_{j-1}}\right)^{1/2}\frac{1}{\overline{\hat{\alpha}_{j-1}}}\right),$$

where now $\delta_j = \langle B\hat{p}_j, \hat{p}_j\rangle$. Thus, regardless of the algorithm used, the matrix H_k can always be constructed without additional inner products.

6.3. CG(B, C, A) as a Lanczos procedure. We now investigate the relationship between the spectra of $\tilde{T}_k$ and CA. Let $\{\tau_{kj}\}_{j=1}^k$ be the eigenvalues of T_k. These are easily found from the eigenvalues, $\{\eta_{kj}\}$, of H_k:

$$\tau_{kj} = e^{i\theta}\left(i\frac{r}{2} + \eta_{kj}\right), \quad j = 1, \cdots, k.$$

Since H_k is Hermitian, each η_{kj} is real. Also notice that H_k is a leading principle submatrix of H_d, where $d = d(s_0, CA)$. (Recall that CG converges to $x = A^{-1}b$ in at most d steps.) Since H_d is tridiagonal, the eigenvalues of H_k *interlace* those of H_{k+1}:

$$\eta_{k+1,j} \le \eta_{kj} \le \eta_{k+1,\,j+1}. \tag{6.6}$$

See [30]. Therefore, the eigenvalues of $\tilde{T}_k$ interlace those of $\tilde{T}_{k+1}$ along $H(CA)$. Finally, let $\sigma_d(CA)$ be the spectrum of $\tilde{T}_d$. This is the set of eigenvalues of CA corresponding to eigenvectors present in the eigenvector expansion of s_0. By the interlacing property above, the eigenvalues of $\tilde{T}_k$ converge to $\sigma_d(CA)$ as $k \to d$. Consequently, CG(B, C, A) may be viewed as a generalized Lanczos procedure.

6.4. Estimating $\kappa_B(CA)$. When BCA is definite, it is possible to estimate $\kappa_B(CA) = \|CA\|_B\|(CA)^{-1}\|_B$ from $H(H_k)$. To see how, first note that only the algebraically smallest and largest eigenvalues of H_k, $\eta_{\min}$ and $\eta_{\max}$, are needed to determine $H(H_k)$, and thus to estimate $H(CA)$. In particular,

$$H(H_k) = \{z : z = \alpha\eta_{\min} + (1-\alpha)\eta_{\max},\ \alpha \in [0,1]\}.$$

The corresponding estimate for $H(CA)$ is

$$\{z : z = \alpha\lambda_{\min} + (1-\alpha)\lambda_{\max},\ \alpha \in [0,1]\} \subset H(CA), \tag{6.7}$$

where

$$\lambda_j = e^{i\theta}\left(i\frac{r}{2} + \eta_j\right),\quad j = \min,\max$$

as in equation (6.2). Since the inclusion in (6.7) may be proper, we have

$$\|CA\|_B \geq \max\{|\lambda_{\min}|, |\lambda_{\max}|\}. \tag{6.8}$$

Estimating $\|(CA)^{-1}\|_B$ is trickier. First observe that $H(CA)$ does not include the origin because BCA is definite. If the matrix G in (2.8) is definite, then $\eta_{\min}\eta_{\max} > 0$, and so

$$\|(CA)^{-1}\|_B \geq \max\{|\lambda_{\min}|^{-1}, |\lambda_{\max}|^{-1}\}. \tag{6.9}$$

If G is indefinite, then $\eta_{\min}\eta_{\max} \leq 0$, in which case it is difficult to estimate the smallest eigenvalue of CA. However, we do have

$$\|(CA)^{-1}\|_B \leq \frac{2}{r}. \tag{6.10}$$

Combining (6.8) and (6.9) or (6.8) and (6.10) yields an estimate for $\kappa_B(CA)$. Note that successive estimates for $\kappa_B(CA)$ are nondecreasing because of the interlacing property (6.6).

When BCA is indefinite, $r = 0$ and $\eta_{\min}\eta_{\max} \leq 0$. Equation (6.8) still holds, but an estimate for $\|(CA)^{-1}\|_B$ is impossible. Also note that the results above hold for any B for which CA is B-normal(1). For example, in GCGHS the matrix CA is both CA-normal(1) and I-normal(1). This will be important in the stopping criteria discussed next.

6.5. Stopping criteria. In any iterative algorithm an important practical question is this: when should the algorithm halt? Most algorithms halt when the relative error is sufficiently small in some norm, that is, when

$$\frac{\|e_i\|_M}{\|x\|_M} \leq \epsilon \tag{6.11}$$

for some hpd M and $\epsilon > 0$. There are many norms in which to measure the relative error. For example, the residual norm ($M = A^*A$) yields the stopping criterion

$$\frac{\|e_i\|_M}{\|x\|_M} = \frac{\|r_i\|}{\|b\|} \leq \epsilon \tag{6.12}$$

where $\|\cdot\|$ denotes the Euclidean norm. If $M = (CA)^*(CA)$, the preconditioned residual norm results, and it gives

$$\frac{\|e_i\|_M}{\|x\|_M} = \frac{\|s_i\|}{\|Cb\|} \leq \epsilon \tag{6.13}$$

where $s_i = Cr_i$ is the preconditioned residual. Finally, if C is hpd, $M = A^*CA$ yields

$$\frac{\|e_i\|_M}{\|x\|_M} = \left(\frac{\langle s_i, r_i\rangle}{\langle Cb, b\rangle}\right)^{1/2} \leq \epsilon. \tag{6.14}$$

Note that each stopping criterion above is computable because r_i and s_i are computable. However, its evaluation may require computing additional inner products, thus increasing the expense of the algorithm. If Omin is used, the following idea of Joubert [26] avoids this extra expense until convergence is near. To explain, suppose the stopping criterion (6.12) is desired in GCGHS, in which $\|s_i\|$ is computed but $\|r_i\|$ is not. If $\|C\|$ is known, we may use the inequality $\|s_i\| \leq \|C\|\,\|r_i\|$ to avoid computing $\|r_i\|$. Specifically, if

$$\frac{\|s_i\|}{\|C\|\,\|b\|} > \epsilon, \tag{6.15}$$

then (6.12) cannot be satisfied. Thus $\|r_i\|$ need not be computed until (6.15) is violated. Similar results hold for other methods.

If an a priori estimate for the M-condition number of CA is available, we may use the inequality

$$\frac{\|e_i\|_M}{\|x\|_M} \leq \kappa_M(CA)\frac{\|s_i\|_M}{\|Cb\|_M} \tag{6.16}$$

to implement a stopping criterion based on the M-norm, even if this norm is uncomputable. For example, suppose $M = I$. If the algorithm is halted when the right-hand side in (6.16) is less than or equal to ϵ, then the relative error in the computed solution is at most ϵ. Unfortunately, $\kappa_M(CA)$ is seldom known a priori. However, if CA is definite, we may *dynamically* estimate $\kappa_B(CA)$ from the eigenvalue estimates derived above; recall equations (6.8)–(6.10). Since CA is definite, the Omin algorithm should be used, in which case the quantity $\langle C^*Be_i, r_i\rangle$ is available from the numerator of $\hat{\alpha}_i$. Thus the stopping criterion

$$\frac{\|e_i\|_B}{\|x\|_B} \leq \left(\kappa_B(CA)\frac{|\langle C^*Be_i, r_i\rangle|}{|\langle C^*Bx, b\rangle|}\right)^{1/2} \leq \epsilon \tag{6.17}$$

is easily and cheaply implemented. Its satisfaction guarantees that the B-norm of the relative error is at most ϵ. Note that C^*Bx is computable whenever C^*Be_i is computable, and so (6.17) is computable.

If $\kappa_I(B)$ is available, (6.17) and

$$\frac{\|e_i\|}{\|x\|} \leq \sqrt{\kappa_I(B)}\frac{\|e_i\|_B}{\|x\|_B}$$

yield

$$\frac{\|e_i\|}{\|x\|} \leq \left(\kappa_I(B)\kappa_B(CA)\frac{|\langle C^*Be_i, r_i\rangle|}{|\langle C^*Bx, b\rangle|}\right)^{1/2} \leq \epsilon, \tag{6.18}$$

which is a bound on the relative error in the I-norm. The availability of $\kappa_I(B)$ depends on the pattern being used; see [3].

Recall that the eigenvalue estimates obtained above yield an approximation to $\kappa_B(CA)$ only when CA is definite. Moreover, this approximation need not be updated (increased) at each step. Instead, it should be updated when the stopping criterion using the current approximation is satisfied. For example, if (6.17) is used, the approximation for $\kappa_B(CA)$ should be updated only when the second inequality in (6.17) is satisfied. If this inequality still holds using the new approximation for $\kappa_B(CA)$, then the iteration should be stopped.

7. Summary. This paper provides a framework for the analysis and classification of conjugate gradient methods. Specifically, any 3-term CG method for $Ax = b$ is characterized by an hpd inner product matrix B and a left preconditioning matrix C. The algorithm Odir(B, C, A) is presented and shown to converge to $x = A^{-1}b$ in at most $d(s_0, CA)$ steps for every x_0 if and only if CA is B-normal(1) or $d(CA) \leq 3$. When BCA is definite, the more efficient Omin(B, C, A) has a similar property. The characterization above leads to a taxonomy for CG methods, which is used to classify and investigate a variety of methods. Finally, eigenvalue estimates for CA are computed from a Hermitian tridiagonal matrix constructed from the CG iteration parameters. These estimates may be used to implement a stopping criterion based more nearly on the true error rather than residual error.

Acknowledgments. We thank Wayne Joubert and David Young for their keen interest and helpful discussions. We also thank G. Minerbo, A. Peterson, C. Smith, and D. Tanner for their persistent interest and probing questions. We are appreciative of the referees' careful reading and many constructive comments.

REFERENCES

[1] L. M. Adams, *Iterative Algorithms for Large, Sparse Linear Systems on Parallel Computers*, Ph.D. thesis, Dept. of Applied Mathematics, University of Virginia, 1982.

[2] S. F. Ashby, *Polynomial Preconditioning for Conjugate Gradient Methods*, Ph.D. thesis, Tech. Report 1355, Dept. of Computer Science, University of Illinois, Urbana, December 1987.

[3] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, Tech. Report UCRL-98508, Lawrence Livermore National Laboratory, Livermore, CA, March 1988.

[4] ———, *Adaptive polynomial preconditioning for Hermitian indefinite linear systems*, BIT, 29 (1989), pp. 583–609.

[5] T. F. Chan, C. J. Kuo, and C. Tong, *Parallel elliptic preconditioners: Fourier analysis and performance on the Connection Machine*, Tech. Report CAM 88-22, Dept. of Mathematics, University of California, Los Angeles, 1988.

[6] R. Chandra, *Conjugate Gradient Methods for Partial Differential Equations*, Ph.D. thesis, Tech. Report 129, Dept. of Computer Science, Yale University, New Haven, CT, 1978.

[7] R. Chandra, S. C. Eisenstat, and M. H. Schultz, *The modified conjugate residual method for partial differential equations*, in Advances in Computer Methods for Partial Differential Equations II, R. Vichnevetsky, ed., 1977, pp. 13–19.

[8] R. C. Chin, T. A. Manteuffel, and J. de Pillis, *ADI as a preconditioning for solving the convection-diffusion equation*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 281–299.

[9] P. Concus and G. H. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, Tech. Report 535, Computer Science Dept., Stanford University, 1976.

[10] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309–332.

[11] E. J. Craig, *The N-step iteration procedures*, J. Math. Physics, 34 (1955), pp. 64–73.

[12] J. E. DENNIS, JR. AND K. TURNER, *Generalized conjugate directions*, Linear Algebra Appl., 88/89 (1987), pp. 187–209.
[13] P. F. DUBOIS, A. GREENBAUM, AND G. H. RODRIGUE, *Approximating the inverse of a matrix for use on iterative algorithms on vector processors*, Computing, 22 (1979), pp. 257–268.
[14] S. C. EISENSTAT, *A note on the generalized conjugate gradient method*, SIAM J. Numer. Anal., 20 (1983), pp. 358–361.
[15] H. C. ELMAN, *Iterative Methods for Large, Sparse Nonsymmetric Systems of Linear Equations*, Ph.D. thesis, Tech. Report 229, Dept. of Computer Science, Yale University, New Haven, CT, 1982.
[16] V. FABER AND T. A. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[17] ———, *Orthogonal error methods*, SIAM J. Numer. Anal., 24 (1987), pp. 170–187.
[18] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.
[19] L. A. HAGEMAN, F. T. LUK, AND D. M. YOUNG, *On the equivalence of certain iterative acceleration methods*, SIAM J. Numer. Anal., 17 (1980), pp. 852–873.
[20] M. R. HESTENES, *The conjugate-gradient method for solving linear systems*, in Proceedings of the Symposium in Applied Mathematics, Vol. 6, New York, 1956, American Mathematical Society, pp. 83–102.
[21] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–435.
[22] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.
[23] K. C. JEA AND D. M. YOUNG, *On the simplification of generalized conjugate-gradient methods for nonsymmetrizable linear systems*, Linear Algebra Appl., 52/53 (1983), pp. 399–417.
[24] O. G. JOHNSON, C. A. MICCHELLI, AND G. PAUL, *Polynomial preconditioning for conjugate gradient calculations*, SIAM J. Numer. Anal., 20 (1983), pp. 362–376.
[25] T. L. JORDAN, *Conjugate gradient preconditioners for vector and parallel processors*, in Proceedings of the Conference on Elliptic Problem Solvers, G. Birkhoff and A. Schoenstadt, eds., New York, 1984, Academic Press.
[26] W. D. JOUBERT, *Private communication*, 1987.
[27] W. D. JOUBERT AND D. M. YOUNG, *Necessary and sufficient conditions for the simplification of generalized conjugate gradient algorithms*, Linear Algebra Appl., 88/89 (1987), pp. 449–485.
[28] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.
[29] P. D. MEYER, A. J. VALOCCHI, S. F. ASHBY, AND P. E. SAYLOR, *A numerical investigation of the conjugate gradient method as applied to three-dimensional groundwater flow problems in randomly heterogeneous porous media*, Water Resources Research, 25 (1989), pp. 1440–1446.
[30] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[31] H. RUTISHAUSER, *Theory of gradient methods*, Mitt. Angew. Math., 8 (1959), pp. 24–49.
[32] Y. SAAD AND M. H. SCHULTZ, *Conjugate gradient-like algorithms for solving nonsymmetric linear systems*, Math. Comp., 44 (1985), pp. 417–424.
[33] ———, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[34] O. WIDLUND, *A Lanczos method for a class of nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–811.
[35] D. M. YOUNG AND K. C. JEA, *Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.