A Unified Approach to Short Recurrence Preconditioned Conjugate Gradient-Type Methods *

A. Mohsen and R. Weiner

Abstract

The problem of iteratively solving a finite real system of linear equations is addressed. Functionals depending on the residual are introduced. The minimization of these functionals for different search direction updates yields several mathematically equivalent, but computationally different, CG-types satisfying certain orthogonality relations. The presented forms reduce to known CG-type methods. Generalizations of Rutishauser's three-term relation and Lanczos three term forms are given.

Key words. CG-type methods, iteration methods
AMS(MOS) subject classifications. 65F10, 65H10
Short title: CG-type methods

1 Introduction

We consider the solution of the linear system

$$Ax = b, \tag{1.1}$$

where A is a given real square nonsingular matrix of order N and b is a given column vector. These systems arise very frequently in scientific computing, for example, in finite difference or finite element approximation of PDEs and as intermediate step in linear and nonlinear programming.
One means of solving (1.1) is by the use of direct methods which are generally variants of Gauss elimination. Direct methods are able to yield the exact solution to (1.1) in a finite number of steps in exact arithmetic. However, as A becomes large (or close to ill-conditioned) direct methods typically become very expensive (or sometimes useless). When A is both large and sparse, iterative methods become an important alternative.
The CG-method was first developed in the late 1940's and early 1950's. The original method [9] is effective when A is symmetric positive definite (spd). The popularity of the method is due to several strong points including

*The first author acknowledges with sincere appreciation the financial support provided by DFG and the working environment facilitated by the Inst. of Numerical Math., University of Halle.

- a short recurrence relation which makes the execution time per iteration and the memory requirements acceptable,
- a finite termination property. In particular if A has clustered eigenvalues, the solution may be reached in few iterations,
- ability to treat ill-conditioned systems,
- use of approximate solution and ability of termination when an "acceptable" accuracy is reached which further reduces computational efforts,
- unlike many iterative methods, CG does not require any estimation of parameters,
- computation is dominated by matrix vector multiplications which fully exploit the sparsity structure of A beside being convenient for parallelization.

The original CG-method may also be effective when A is symmetrizable. In this case there exists a spd matrix Z such that ZA is spd. A straightforward extension to a general nonsymmetric matrix is to apply CG to either one of the spd systems [9]

$$A^T A x = A^T b \qquad\qquad \text{[CGNR]}$$

or [3]

$$AA^T y = b, \quad x = A^T y, \qquad \text{[CGNE]}$$

where A^T is the transpose of A. Although there are special situations where CGNR or CGNE are optimal extensions of CG, both algorithms may converge slowly and hence they may not be satisfactory generalizations of CG to arbitrary nonsymmetric matrices.
There has been extensive search for generalized CG-methods which can be applied to nonsymmetric, indefinite and/or inconsistent problems and which possess some of its appealing properties. This search led to a wide variety of methods which possess long recurrence relations making them expensive in the general case. Limited recurrence with restart are usually recommended [1], [15].
In order to improve the speed of convergence of iterative methods, one usually applies some form of preconditioning. Several preconditioners have been suggested over the years [1], [15], [7]. Preconditioning must be regarded as a trade off between the cost of constructing and manipulating the preconditioner and the acceleration of the iterative process.
Interest in short-recurrence CG-type methods is natural due to their relative simplicity, low computational cost and storage requirements per iteration. They are applicable at least when A is spd or if it is symmetrizable as well as when the normal equations have to be solved or can be economically solved for A of low condition number. The scope of applicability is widen by the growing interest and success in using preconditioning. Such an interest stimulated our present search in how the known relations in short-recurrence CG-type methods apply under preconditioning. In this work a unified treatment of classical short term CG-type methods, which also takes preconditioning into account, is presented. Orthogonality conditions and several other properties are proved. Generalizations of Rutishauser's three term relation [14] and Lanczos three term form [11] are given. A previous related study using a different approach was done in [19].

2 Basics

Let the residual at the i-th iteration be defined by

$$r_i = b - Ax_i.$$

The updated solution is given by

$$x_{i+1} = x_i + \alpha_i p_i, \tag{2.1}$$

where p_i is the i-th search direction. It follows that

$$r_{i+1} = r_i - \alpha_i A p_i. \tag{2.2}$$

Here α_i is defined such that the functional

$$F_{i+1} = (r_{i+1}, S r_{i+1}) \tag{2.3}$$

is minimized. S is a symmetric positive definite (spd) matrix and

$$(u, v) = u^T v.$$

Similar functionals were introduced previously [10], [12], [16] but for different purposes. Introducing r_{i+1} in (2.3) and putting $\partial F_{i+1}/\partial \alpha_i = 0$, we get

$$\alpha_i = (r_i, SAp_i)/(Ap_i, SAp_i) \tag{2.4}$$

which, using (2.2), implies that

$$(r_{i+1}, SAp_i) = 0. \tag{2.5}$$

For this value of α_i, we have

$$F_{i+1} = F_i - \alpha_i^2 (Ap_i, SAp_i) \leq F_i$$

which shows the reduction in F_i with iteration.
Now let the updated search directions be defined by

$$p_{i+1} = g_{i+1} + \beta_i p_i \tag{2.6}$$

where g_{i+1} is a suitable chosen vector which is linearly dependent on r_{i+1}, i.e.

$$g_{i+1} = Q r_{i+1},$$

where Q represents a suitable preconditioner, i.e., we consider the preconditioned system

$$QAx = Qb.$$

We set $p_0 = Q r_0$. Imposing the conjugacy requirement

$$(Ap_{i+1}, SAp_i) = 0 \tag{2.7}$$

we have

$$\beta_i = -(Ag_{i+1}, SAp_i)/(Ap_i, SAp_i). \tag{2.8}$$

It follows from (2.2), (2.3) and (2.6)that

$$\begin{aligned} F_{i+1} =& (r_i - \alpha_i Ag_i, S[r_i - \alpha_i Ag_i]) - 2\alpha_i\beta_{i-1}(r_i, SAp_{i-1}) \\ &+ 2\alpha_i^2\beta_{i-1}(Ag_i, SAp_{i-1}) + \alpha_i^2\beta_{i-1}^2(Ap_{i-1}, SAp_{i-1}). \end{aligned}$$

The coefficient of $\alpha_i\beta_{i-1}$ is zero due to (2.5), hence β_i given by (2.8) also minimizes the functional.

3 Further relations

We next prove the relations

$$\begin{aligned} (r_i, SAp_i) &= (r_i, SAg_i) && (3.1)\\ (Ap_i, SAp_i) &= (Ag_i, SAp_i) && (3.2)\\ (Ap_i, SAp_i) &\leq (Ag_i, SAg_i) && (3.3)\\ (r_i, SAp_j) &= 0 \quad \text{for } i > j && (3.4)\\ (r_i, SAg_j) &= 0 \quad \text{for } i > j && (3.5)\\ (Ap_i, SAp_j) &= 0 \quad \text{for } i \neq j && (3.6)\\ (Ap_i, SAg_j) &= 0 \quad \text{for } i \neq j,\ i \neq j-1 && (3.7)\\ (r_i, SAp_j) &= (r_0, SAp_j) \quad \text{for } i \leq j && (3.8) \end{aligned}$$

These forms generalize previously known relations, e.g.[13], [4] which were derived for particular S and g and no preconditioning was considered.
Using (2.5) and (2.6) it follows that

$$\begin{aligned} p_i &= g_i + \beta_{i-1}p_{i-1} \\ (r_i, SAp_i) &= (r_i, SAg_i) + \beta_{i-1}(r_i, SAp_{i-1}) = (r_i, SAg_i), \end{aligned}$$

i.e. (3.1). Starting with eq. (2.5) and using (2.7) we have

$$(Ap_i, SAp_i) = (Ag_i, SAp_i) + \beta_{i-1}(Ap_{i-1}, SAp_i) = (Ag_i, SAp_i),$$

i.e. (3.2). To prove (3.3) we have

$$\begin{aligned} (Ap_i, SAp_i) &= (Ag_i + \beta_{i-1}Ap_{i-1}, SAg_i + \beta_{i-1}SAp_{i-1}) \\ &= (Ag_i, SAg_i) + 2\beta_{i-1}(Ag_i, SAp_{i-1}) + \beta_{i-1}^2(Ap_{i-1}, SAp_{i-1}) \\ &= (Ag_i, SAg_i) - \frac{(Ag_i, SAp_{i-1})^2}{(Ap_i, SAp_i)} \end{aligned}$$

where eq. (2.8) is employed. Since the second term is always positive, (3.3) follows directly.

We follow Reid [13] to prove (3.4)–(3.6) jointly. From (2.5) and (2.7) it follows that (3.4) and (3.6) are valid for $i = j + 1$. From eq. (2.2), we have

$$(r_{i+1}, SAg_i) = (r_i, SAg_i) - \alpha_i(Ap_i, SAg_i) = 0$$

from (3.1), (3.2) and using the definition of α_i. Thus (3.5) is valid for $i = j + 1$. Now by induction, we assume that (3.4)–(3.6) are satisfied for $i = j + 2$ up to k. Then using (3.3) we have

$$(r_{k+1}, SAp_j) = (r_k, SAp_j) - \alpha_k(Ap_k, SAp_j)$$

which is zero via the induction assumptions and so (3.4) is satisfied. Now using (2.6), we have

$$g_j = p_j - \beta_{j-1}p_{j-1}$$
$$(r_{k+1}, SAg_j) = (r_{k+1}, SAp_j) - \beta_{j-1}(r_{k+1}, SAp_{j-1})$$

which is zero based on (3.4) and hence (3.5) is satisfied. Using eq. (2.6) again we have

$$(Ap_{k+1}, SAp_j) = \frac{1}{\alpha_{k+1}}(r_k - r_{k+1}, SAp_j) = 0$$

based on (3.4). Thus we have

$$(Ap_i, SAp_j) = 0 \quad \text{for } i > j.$$

Since S is spd, it follows that

$$(Ap_i, SAp_j) = 0 \quad \text{for } i \neq j.$$

Starting with eq. (2.6) and using (3.6), we may derive (3.7) directly.
Relation (3.8) is proved by induction. It is trivially true for $i = 0$. Assume that it holds for $i < k < j$, then using (3.3) we have

$$\begin{aligned}(r_{k+1}, SAp_j) &= (r_k, SAp_j) - \alpha_k(Ap_k, SAp_j)\\ &= (r_k, SAp_j) = (r_{k-1}, SAp_j) = ... = (r_0, SAp_j).\end{aligned}$$

The described algorithm allows to give an approximation for the inverse of A. Starting with $x_0 = 0$ we have a numerical solution

$$x = \sum_{l=1}^{m} \alpha_l p_l,$$

where m is the number of search directions included. This start is not a restriction as $Ax = b$ could be written as

$$A(x - x_0) = b - Ax_0.$$

Upon application of the conjugacy relation, we get

$$x = \sum_{l=1}^{m} \frac{(b, SAp_l)}{(Ap_l, SAp_l)} p_l$$

which shows that the elements of the inverse of A are approximately given by

$$a_{ij}^{(-1)} = \sum_{l=1}^{m} \frac{1}{(Ap_l, SAp_l)} p_{li} (SAp_l)_j.$$

Following [4] we next prove that

$$\text{span}\{p_0, p_1, ..., p_i\} = \text{span}\{p_0, QAp_0, ..., (QA)^i p_0\} = \text{span}\{g_0, g_1, ..., g_i\}.$$

The three spaces are identical when $i = 0$. Assume that the statement is true for $i \leq k$. By (2.6) it follows immediately that

$$\text{span}\{p_0, ..., p_{k+1}\} = \text{span}\{g_0, ..., g_{k+1}\}.$$

On the other hand using

$$g_{k+1} = Qr_{k+1} = g_k - \alpha_k QAp_k$$

we may write

$$p_{k+1} = g_k - \alpha_k QAp_k + \beta_k p_k,$$

from which we deduce that

$$\text{span}\{p_0, ..., p_{k+1}\} = \text{span}\{p_0, ..., (QA)^{k+1} p_0\}.$$

4 Special considerations

The above relations yield several forms of α_i and β_i. In particular, we have

$$\begin{aligned}\alpha_i &= \frac{(r_i, SAp_i)}{(Ap_i, SAp_i)} = \frac{(r_0, SAp_i)}{(Ap_i, SAp_i)} = \frac{(r_i, SAp_i)}{(Ag_i, SAp_i)} = \frac{(r_0, SAp_i)}{(Ag_i, SAp_i)} \\ &= \frac{(r_i, SAg_i)}{(Ap_i, SAp_i)} = \frac{(r_i, SAg_i)}{(Ap_i, SAg_i)}\end{aligned}$$

$$\begin{aligned}\beta_i &= -\frac{(Ag_{i+1}, SAp_i)}{(Ap_i, SAp_i)} = -\frac{(Ag_{i+1}, Sr_{i+1} - Sr_i)}{(Ap_i, Sr_{i+1} - Sr_i)} = \frac{(Ag_{i+1}, Sr_{i+1} - Sr_i)}{(Ap_i, Sr_i)} = \frac{(Ag_{i+1}, Sr_{i+1} - Sr_i)}{(r_0, SAp_i)} \\ &= -\frac{(Ag_{i+1}, SAp_i)}{(Ag_i, SAp_i)} = -\frac{(Ag_{i+1}, Sr_{i+1} - Sr_i)}{(Ag_i, Sr_{i+1} - Sr_i)} = \frac{(Ag_{i+1}, Sr_{i+1} - Sr_i)}{(Ag_i, Sr_i)}\end{aligned}$$

Besides, different combinations of α_i and β_i may be utilized. The particular choice may depend on the characteristics of A, the corresponding appropriate choices of S and g, the computation involved, particularly matrix-vector-multiplication, vector updates, inner products, the resulting round-off error as well as favorable properties on parallel machines.
Next we consider special choices of S and g depending on whether A is symmetric or not.

4.1 A is spd

The case of a symmetric, positive definite matrix corresponds to the classical CG algorithm. We may let

$$S = A^{-1}.$$

We obtain

$$F_i = (r_i, A^{-1}r_i) = (r_i, A^{-1}(b - Ax_i)) = (r_i, x_e - x_i) = (r_i, \varepsilon_i),$$

where $\varepsilon_i = x_e - x_i$ is the error in the solution and x_e is the exact solution. The reduction in F_i is given by

$$\triangle F_i = F_i - F_{i+1} = \alpha_i^2 (p_i, Ap_i).$$

In this case we have the following relations

$$\begin{aligned}
(SAp_i, Ap_j) &= (p_i, Ap_j) = 0 \quad \text{for } i \neq j \\
(r_i, SAp_i) &= (r_i, p_i) = (r_0, p_i) = (r_i, SAg_i) = (r_i, g_i) \\
(Ap_i, SAp_i) &= (Ap_i, p_i) = (Ap_i, SAg_i) = (g_i, Ap_i) = (p_i, Ag_i) \\
(Ag_{i+1}, SAp_i) &= (Ag_{i+1}, p_i) = (g_{i+1}, Ap_i)
\end{aligned}$$

Since the numerator and denominator in α_i and β_i take different forms, any combinations may be valid. However, for β_i we may introduce further relations via simultaneous substitution of p_i or Ap_i in the numerator and denominator in terms of $(x_{i+1} - x_i)$ or $(r_{i+1} - r_i)$, respectively. Thus we have

$$\begin{aligned}
\beta_i &= -\frac{(Ag_{i+1}, p_i)}{(Ap_i, p_i)} = -\frac{(Ag_{i+1}, x_{i+1} - x_i)}{(Ap_i, x_{i+1} - x_i)} \\
&= -\frac{(g_{i+1}, Ap_i)}{(p_i, Ap_i)} = -\frac{(g_{i+1}, r_{i+1} - r_i)}{(p_i, r_{i+1} - r_i)} = \frac{(g_{i+1}, r_{i+1} - r_i)}{(p_i, r_i)} = \frac{(g_{i+1}, r_{i+1} - r_i)}{(g_i, r_i)}
\end{aligned}$$

where orthogonality relations have been utilized. We note that these expressions yield the known forms taking the possibility of preconditioning. Advantages of particular expressions compared to others were recognized by Hestenes and Stiefel [9] as well as Reid [13] and Stoer [17]. We note that in this case

$$g_j = Qr_j$$

where Q is spd. Consequently eq. (3.5) yields

$$(r_i, g_j) = 0 \quad \text{for } i \neq j,$$

resulting in

$$\beta_i = \frac{(g_{i+1}, r_{i+1})}{(g_i, r_i)}.$$

4.2 General matrix

I) For the trivial choice $S = I$ we have

$$F_i = \|r_i\|^2$$
$$\triangle F_i = \alpha_i^2 \|Ap_i\|^2$$

Some of the relations are

$$\begin{aligned}
(Ap_i, Ap_j) &= 0 \quad \text{for } i \neq j \\
(r_i, SAp_i) &= (r_i, Ap_i) = (A^T r_i, p_i) = (r_0, Ap_i) \\
&= (r_i, SAg_i) = (r_i, Ag_i) = (A^T r_i, g_i) \\
(Ap_i, SAp_i) &= (Ap_i, Ap_i) = (Ap_i, SAg_i) = (Ap_i, Ag_i) \\
(Ag_{i+1}, SAp_i) &= (Ag_{i+1}, Ap_i) \\
\beta_i &= -\frac{(Ag_{i+1}, Ap_i)}{(Ap_i, Ap_i)} = -\frac{(Ag_{i+1}, r_{i+1} - r_i)}{(Ap_i, r_{i+1} - r_i)} = \frac{(Ag_{i+1}, r_{i+1} - r_i)}{(Ap_i, r_i)} \\
&= \frac{(Ag_{i+1}, r_{i+1} - r_i)}{(Ag_i, r_i)} = \frac{(g_{i+1}, A^T(r_{i+1} - r_i))}{(g_i, A^T r_i)}
\end{aligned}$$

We may use

$$Ap_{i+1} = Ag_{i+1} + \beta_i Ap_i$$

to find Ap_{i+1} once Ag_{i+1} is computed. This case corresponds to the conjugate residual (CR) method [4] and reduces to formulas given in [9] for CGNR upon putting $g = A^T r$.
II) If we let

$$S = (AA^T)^{-1} \quad \text{and} \quad g_i = A^T \tilde{Q} r_i = A^T v_i$$

we have

$$\begin{aligned}
F_i &= (r_i, (AA^T)^{-1} r_i) = \|x_e - x_i\|^2 \\
\triangle F_i &= \alpha_i^2 \|p_i\|^2 \\
(p_i, p_j) &= 0 \quad \text{for } i \neq j \\
(r_i, SAp_i) &= (r_i, SAg_i) = (r_i, v_i) \\
(Ap_i, SAp_i) &= (p_i, p_i) = (Ag_i, SAp_i) = (v_i, Ap_i) = (g_i, p_i) \\
(Ag_{i+1}, SAp_i) &= (v_{i+1}, Ap_i) = (g_{i+1}, p_i) \\
\beta_i &= -\frac{(v_{i+1}, Ap_i)}{(v_i, Ap_i)} = -\frac{(v_{i+1}, r_{i+1} - r_i)}{(v_i, r_{i+1} - r_i)} = \frac{(v_{i+1}, r_{i+1} - r_i)}{(v_i, r_i)} \\
&= -\frac{(g_{i+1}, p_i)}{(g_i, p_i)} = -\frac{(g_{i+1}, x_{i+1} - x_i)}{(g_i, x_{i+1} - x_i)}
\end{aligned}$$

This case is related to the work of Craig [3] taking preconditioning into consideration.

5 Derivations of three-term relations

Next we derive three-term relations for updating $\{x_i\}$ as well as for updating $\{p_i\}$.
We may update $\{x_i\}$ after eliminating $\{p_i\}$ using (2.1) and (2.6):

$$\begin{aligned} x_{i+1} &= x_i + \alpha_i(g_i + \beta_{i-1}p_{i-1}) \\ &= x_i + \alpha_i g_i + \frac{\alpha_i}{\alpha_{i-1}}\beta_{i-1}(x_i - x_{i-1}). \end{aligned}$$

It also follows that

$$r_{i+1} = r_i - \alpha_i A g_i + \frac{\alpha_i}{\alpha_{i-1}}\beta_{i-1}(r_i - r_{i-1}). \tag{5.1}$$

Elimination of $\{p_i\}$ from the factors multiplying g_i and $(x_i - x_{i-1})$ in (5.1) requires some manipulations. We write

$$x_{i+1} = x_i + \frac{1}{q_i}[g_i + e_{i-1}(x_i - x_{i-1})]$$

where

$$q_i = \frac{1}{\alpha_i}, \qquad e_i = \frac{\beta_i}{\alpha_i} = \beta_i q_i.$$

But we have

$$\begin{aligned} \frac{(Ag_i, SAg_i)}{(r_i, SAg_i)} &= \frac{1}{(r_i, SAg_i)}(A(p_i - \beta_{i-1}p_{i-1}), SA(p_i - \beta_{i-1}p_{i-1})) \\ &= \frac{1}{(r_i, SAg_i)}[(Ap_i, SAp_i) + \beta_{i-1}^2(Ap_{i-1}, SAp_{i-1})] \\ &= \frac{1}{\alpha_i} + \frac{\beta_{i-1}^2}{(r_i, SAg_i)}[-\frac{1}{\beta_{i-1}}(Ag_i, SAp_{i-1})] \\ &= \frac{1}{\alpha_i} + \frac{\beta_{i-1}}{\alpha_{i-1}}\frac{(Ag_i, Sr_i - Sr_{i-1})}{(r_i, SAg_i)} \\ &= \frac{1}{\alpha_i} + e_{i-1}\frac{(r_i - r_{i-1}, SAg_i)}{(r_i, SAg_i)}, \end{aligned}$$

consequently

$$\begin{aligned} q_i &= \frac{(Ag_i, SAg_i)}{(r_i, SAg_i)} - e_{i-1}\frac{(r_i - r_{i-1}, SAg_i)}{(r_i, SAg_i)}, \quad e_{-1} = 0, \\ e_i &= q_i\frac{(r_{i+1} - r_i, SAg_{i+1})}{(r_i, SAg_i)}. \end{aligned}$$

So we have an updating formula for x_{i+1} independent of $\{p_i\}$. This is a generalization of Rutishauser's three-term relation [14] and may be related to the generalized CG [2] upon taking $S = I$.
We may update $\{p_i\}$ such that the expressions do not involve $\{g_i\}$.

Let $g_{i+1} = Qr_{i+1}$. In this case we have

$$\begin{aligned} p_{i+1} &= Qr_{i+1} + \beta_i p_i \\ &= Qr_i - \alpha_i QAp_i + \beta_i p_i \\ &= (p_i - \beta_{i-1}p_{i-1}) - \alpha_i QAp_i + \beta_i p_i \end{aligned}$$

This is equivalent to the scaled form

$$\bar{p}_{i+1} = QA\bar{p}_i - \bar{\alpha}_i \bar{p}_i - \bar{\beta}_{i-1}\bar{p}_{i-1}.$$

Upon application of (3.6), we get

$$\begin{aligned} \bar{\alpha}_i &= (AQA\bar{p}_i, SA\bar{p}_i)/(A\bar{p}_i, SA\bar{p}_i) \\ \bar{\beta}_{i-1} &= (AQA\bar{p}_i, SA\bar{p}_{i-1})/(A\bar{p}_{i-1}, SA\bar{p}_{i-1}) \end{aligned}$$

These are generalizations of Lanczos expression which reduce to
(I) For A spd and $S = A^{-1}$:

$$\begin{aligned} \bar{\alpha}_i &= (QA\bar{p}_i, A\bar{p}_i)/(\bar{p}_i, A\bar{p}_i) \\ \bar{\beta}_{i-1} &= (QA\bar{p}_i, A\bar{p}_{i-1})/(\bar{p}_{i-1}, A\bar{p}_{i-1}) \end{aligned}$$

which agrees with expressions given in Hackbusch [8].
(II) For general A and $S = I$ we have

$$\begin{aligned} \bar{\alpha}_i &= (AQA\bar{p}_i, A\bar{p}_i)/(A\bar{p}_i, A\bar{p}_i) \\ \bar{\beta}_{i-1} &= (AQA\bar{p}_i, A\bar{p}_{i-1})/(A\bar{p}_{i-1}, A\bar{p}_{i-1}), \end{aligned}$$

which corresponds to the stabilized Fridman form [18] in case of preconditioning.
(III) For general A and $S = (A^TA)^{-1}$ we have

$$\begin{aligned} \bar{\alpha}_i &= (QA\bar{p}_i, \bar{p}_i)/(\bar{p}_i, \bar{p}_i) \\ \bar{\beta}_{i-1} &= (QA\bar{p}_i, \bar{p}_{i-1})/(\bar{p}_{i-1}, \bar{p}_{i-1}), \end{aligned}$$

which generalizes Fridman's expression [6] taking preconditioning into account.

6 Discussion

In this work a unified treatment of short-recurrence CG-type methods is presented. The formulation takes possible preconditioning into account. Thus g_i may correspond to a suitable preconditioner Q as $g_i = Qr_i$. Orthogonality relations and several other properties are derived. This leads to several mathematically equivalent, but computationally different CG-types. The choice of a particular form should be based on the type of A and the computational burden encountered. If one does not wish to use any properties of the method in computation of α_i and β_i besides the defining relations, since they may be disturbed by

rounding-off errors, one should use eqs. (2.4) and (2.8). For studies of round off errors in iterative solvers we refer to ([7], Ch 4).
Theoretically the search vectors generated by the algorithm maintain the relation

$$(Ap_i, SAp_j) = 0 \quad i \neq j.$$

However, round-off errors combine to rapidly degrade such orthogonality. One may use the parameter

$$orth = \log_{10} \frac{|(Ap_i, SAp_0)|}{\|Ap_i\| \|SAp_0\|}$$

to detect this loss of orthogonality. If $orth$ rises significantly above a preset value, the algorithm may be restarted. One might also consider forcing orthogonality via a procedure such as Gram-Schmidt. However, this would require the storage of all previously obtained vectors. More economically one may require orthogonality to a limited number of vectors. To avoid possible instability of the orthogonalization process, it is recommended to apply CG algorithms as s-step iterative process. Thus after s iterations we restart with the iterative refinement equation

$$A\triangle x = r.$$

It is also generally advised to use

$$r_i = b - Ax_i$$

occasionally to avoid accumulation of errors. It may be appropriate to refer to the work of Faber and Manteuffel [5] who developed a theory on the existence of CG-type methods with certain desirable features adopted from classical CG and to the discussion in ([7], Ch 6).

References

[1] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*, SIAM, Philadelphia, 1993.

[2] P. Concus and G.H. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, Computing Methods in Applied Sciences and Engineering (R. Glowinski and J.L. Lions, eds.), Lecture Notes in Economics and Mathematical Systems, vol. 134, Springer, Berlin, 1976, pp. 56–65.

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

[4] S.C. Eisenstat, H.C. Elman, and M.H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal. **20** (1983), 345–357.

[5] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal. **21** (1984), 352–362.

[6] V.M. Fridman, *The method of minimum iterations with minimum errors for a system of linear algebraic equations with a symmetric matrix*, USSR Comput. Math. and Math. Phys. **2** (1963), 362–363.

[7] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[8] W. Hackbusch, *Iterative solution of large sparse systems of equations*, Springer, New York, 1994.

[9] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards **49** (1952), 409–436.

[10] B.P. Ilin, *Some estimates for conjugate gradient methods*, USSR Comp. Math. & Math. Phys. **16** (1977), 22–30.

[11] C. Lanczos, *Solutions of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards **49** (1952), 33–53.

[12] A. Meyer, *The concept of special inner product for deriving new conjugate gradient-like solvers for non-symmetric sparse linear systems*, Num. Lin. Alg. Appl. **1** (1994), 129–139.

[13] J.K. Reid, *On the method of conjugate gradients for the solution of large sparse systems of linear equations*, Large sparse sets of linear equations (J.K. Reid, ed.), Academic Press, New York, 1971, pp. 231–254.

[14] H. Rutishauser, *Theory of gradient methods*, Refined Iterative Methods for Computation of the Solution and Eigenvalues of Self-adjoint Boundary Value Problems (M. Engeli, Th. Ginsburg, H. Rutishauser, and E. Stiefel, eds.), Birkhäuser Basel, 1959.

[15] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Company, Boston, 1995.

[16] T.K. Sarkar and E. Ervas, *A survey of various numerical techniques to solve operator equations in electromagnetics*, Annales des Telecomm. **7–8** (1985), 331–340.

[17] J. Stoer, *Solution of large linear systems of equations by conjugate-gradient type methods*, Mathematical Programming: The State of the Art (A. Bachem, M. Grötschel, and B. Korte, eds.), Springer, Berlin, 1983.

[18] J. Stoer and R. Freund, *On the solution of large indefinite systems of linear equations by conjugate gradient algorithms*, Methods in Applied Sciences and Engineering V (R. Glowinski and J.L. Lions, eds.), North Holland, Amsterdam, 1982.

[19] V.V. Voevodin, *Linear Algebra*, Mir Publ., Moscow, 1983.

Adel Mohsen
Eng. Math. Dept.
Eng. Faculty
Cairo University
Giza 12211
Egypt
e-mail: amohsen@alpha1-eng.cairo.eun.eg

Rüdiger Weiner
FB Mathematik und Informatik
Martin-Luther-Universität
Halle-Wittenberg
Postfach
D-06099 Halle/Saale
Germany
e-mail: rw@mail.mathematik.uni-halle.de