NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-627

Sparse Matrix Methods Based on Orthogonality and Conjugacy

C. L. Lawson

(NASA-CR-133231) SPARSE MATRIX METHODS BASED ON ORTHOGONALITY AND CONJUGACY (Jet Propulsion Lab.) 66 p HC $5.50 CSCL 12A

N73-26617

Unclas

G3/19 08275

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1973

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-627

Sparse Matrix Methods Based on Orthogonality and Conjugacy

C. L. Lawson

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1973

PREFACE

The work described in this report was performed by the Data Systems Division of the Jet Propulsion Laboratory.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

ABSTRACT

A matrix having a high percentage of zero elements is called sparse. In the solution of systems of linear equations or linear least squares problems involving large sparse matrices significant saving of computer cost can be achieved by taking advantage of the sparsity. This Memorandum derives and describes the well known conjugate gradient algorithm and a set of related algorithms which are applicable to such problems.

Control of accuracy is a serious problem with this class of methods. We plan to devote a subsequent study to methods of controlling algorithms of this class.

PART I

Introduction

Chapter 1 Introduction

A matrix A is called sparse if a large proportion of its elements are zero. Significant savings of execution time and computer storage can be realized in solving systems of linear equations, $Ax=b$, or least squares problems, $Ax \cong b$, if the matrix A is sparse and if special solution methods are used which take advantage of the sparseness.

Most direct solution methods such as Gaussian elimination or Householder orthogonal triangularization perform transformations on the given matrix A which significantly increase the number of nonzero elements. The two main ideas in developing sparse matrix methods have been

1. Reorganize direct elimination methods to reduce the growth of the number of nonzero elements.

and

2. Use iterative methods so that the original sparse matrix A is used throughout the computation.

In Reid (1971) it is pointed out that the conjugate gradient (CG) method for solving a system $Ax=b$, with A symmetric and positive definite, is well suited for use when A is sparse. This method shares with iterative methods the feature of continually using the initial matrix A but it shares with direct methods the property of theoretically terminating after not more than n iterations.

More specifically the major computational cost in each iteration of the CG method arises from the multiplication of A times a vector. If A is an $n \times n$

matrix and the proportion of nonzero elements in A is ρ then the multiplication of A times an n-vector requires ρn^2 multiplications and additions if multiplication by zero elements of A are skipped. This algorithm theoretically reaches the solution vector in at most n iterations so the number of multiplications and additions would theoretically be at most ρn^3.

The storage required is just that needed for the sparse matrix A plus four n-vectors. Since A is symmetric it has only approximately $\rho n^2/2$ potentially distinct nonzero elements. Such a matrix can be stored in ρn^2 locations or even in less space if the nonzero elements are located in some known regular pattern.

In comparison direct solution of this problem using an efficient stable method such as Cholesky factorization would in general require about $n^2/2$ storage locations and $n^3/6$ multiplications and additions. Thus the CG method will require less storage than the Cholesky method if $\rho < 1/2$ and will require fewer multiplications and additions if $\rho < 1/6$.

These cut-off values for ρ should be taken only as very rough guidelines. Storage management for the sparse matrix method may increase its execution time.

More serious is the lack of numerical stability in the CG method. If the matrix A has a large condition number the intermediate vectors computed by the algorithm which are theoretically orthogonal or A-conjugate (defined in Chapter 2) may not even come close to having these properties. I feel that this means that a reliable subroutine for the CG method must include some monitoring of the error generated in intermediate quantities.

The purpose of this report is to collect in one place, and with some consistency of notation, the statements and theoretical justifications of the conjugate gradient algorithm and a number of other algorithms having very similar characteristics with regard to mathematical theory, operation counts, and storage requirements. We subsequently plan to produce Fortran subroutines for some of these algorithms and study particularly the effectiveness and reliability of various techniques for monitoring accuracy and testing for termination in these subroutines.

The CG algorithm was invented independently and simultaneously (~ 1951) by M. R. Hestenes and E. Stiefel [see Hestenes and Stiefel (1952)]. The paper by Craig (1955) which discusses methods of this type also references related work by Fox, Huskey, Wilkinson, Lanczos, Forsythe, and Rosser in the 1948-1952 period.

After providing some mathematical background in Chapter 2 the CG algorithm is presented in Chapter 3. In Chapters 4 and 5 algorithms are given which result directly from replacing the matrix A in the CG algorithm by A^TA or AA^T respectively. The use of A^TA covers the case of a least squares problem whereas the use of AA^T allows solution of consistent systems, $Ax=b$, in which A is not necessarily symmetric or positive definite (or even square).

This replacement of A by A^TA and by AA^T occurs in Craig (1955) and is also treated in Faddeev and Faddeeva (1963).

More recently Reid (1971) made a strong case for the usefulness of the CG method in large sparse problems. Interest in sparse problems also stimulated Paige (1972) to derive two algorithms of similar character. We will refer to these algorithms as PAIGE-I and PAIGE-II. Paige's derivation of these algorithms is based on a bidiagonalization method given by Golub and Kahan (1965) which has its mathematical roots in a method of Lanczos (1950) for tridiagonalizing a symmetric matrix. We present Paige's least squares algorithm, PAIGE-II, in Chapter 6, where we call it ITLS to distinguish our particular statement of the algorithm. In Chapter 7 we show that ITLS theoretically generates the same sequence of approximate solution vectors as the algorithm of Chapter 4. The intermediate steps are sufficiently different however to make it of interest to investigate the numerical performance of each of these two algorithms.

In Chapter 8 we present an algorithm ITC which is a subset of Paige's algorithm PAIGE-I. The algorithm PAIGE-I includes provision for handling a least squares problem. It appears to me that for least squares problems this algorithm is not competitive with PAIGE-II or the least squares algorithm of Chapter 4 and thus I have presented only the subset of PAIGE-I which handles a consistent system of linear equations. Paige noted that this subset of PAIGE-I is equivalent to the algorithms given by Faddeev and Faddeeva (1963) and by Craig (1955) which are described in Chapter 5 of this report. This equivalence is verified in Chapter 9.

Finally in Chapter 10 we give an algorithm due to C. C. Paige and M. A. Saunders (personal correspondence, 1972) for the symmetric consistent problem, $Ax = b$ This algorithm is called SYMMLQ by Paige and Saunders. Note that whereas the CG algorithm requires only one matrix-vector multiplication per iteration the other algorithms discussed in Chapters 4-9 each require two matrix-vector multiplications per iteration or else the preliminary computation of A^TA or AA^T. The algorithm, SYMMLQ, requires only one matrix-vector multiplication per iteration and thus is nominally the most economical method described in this report for the indefinite symmetric consistent problem. This algorithm is however notably more complicated than the other algorithms in this report.

Saunders has also developed a modification to Paige's least squares algorithm, PAIGE-II, (our Chapter 6) however we have omitted this as it is more complicated than Paige's algorithm and in a few test cases which we ran its sequence of approximate solution vectors was approximately one step behind the sequence generated by Paige's algorithm.

We wish to thank John Reid and Gene Golub for kindling our awareness of the value of this class of methods for sparse problems. We thank Chris Paige and Michael Saunders for sharing their partially completed current research work with us and David Saunders for supplying Fortran implementations of Michael Saunders' two algorithms.

Chapter 2 Mathematical Background

Two real n-vectors x and y are mutually orthogonal if their inner product, denoted by $x^T y$ or $y^T x$, is zero. They are orthonormal if they are mutually orthogonal and are each of unit euclidean length, i.e. $\|x\| \equiv (x^T x)^{1/2} = 1$ and $\|y\| \equiv (y^T y)^{1/2} = 1$.

A generalization of orthogonality is the notion of A-conjugacy where A is a symmetric matrix. Two vectors x and y are A-conjugate if $x^T Ay$ (or equivalently $y^T Ax$) is zero. This notion of A-conjugacy is most commonly defined only for a positive definite symmetric matrix A since then one has the convenient property that $x^T Ax > 0$ for all $x \neq 0$.

We will use the notion of A-conjugacy under the weaker assumption that A is nonnegative definite symmetric matrix but limit the vectors being considered to those lying in the row space of A. For such vectors the property $x^T Ax > 0$ for $x \neq 0$ still holds.

If the set of vectors $\mathcal{V}_i = \{v^{(1)}, \ldots, v^{(i)}\}$ are mutually orthogonal and a vector $y^{(i+1)}$ is linearly independent of the set $\mathcal{V}_i$ then a vector $v^{(i+1)}$ orthogonal to the set $\mathcal{V}_i$ can be defined by

$$(2.1) \quad v^{(i+1)} = y^{(i+1)} - \frac{v^{(1)T} y^{(i+1)}}{v^{(1)T} v^{(1)}} v^{(1)} - \cdots - \frac{v^{(i)T} y^{(i+1)}}{v^{(i)T} v^{(i)}} v^{(i)}$$

This is the formula of Gram-Schmidt orthogonalization.

A similar formula exists for extending a mutually A-conjugate set of vectors. Thus if the set of vectors $\mathcal{U}_i = \{u^{(1)}, \ldots, u^{(i)}\}$ are mutually A-conjugate and a vector $z^{(i+1)}$ is linearly independent of the set $\mathcal{U}_i$ then a vector $u^{(i+1)}$, A-conjugate to $\mathcal{U}_i$, is defined by

$$(2.2) \quad u^{(i+1)} = z^{(i+1)} - \frac{u^{(1)T}Az^{(i+1)}}{u^{(1)T}Au^{(1)}} u^{(1)} - \cdots - \frac{u^{(i)T}Az^{(i+1)}}{u^{(i)T}Au^{(i)}} u^{(i)}$$

if all of the denominators are nonzero. Nonzero denominators are assured if A is positive definite symmetric or if A is nonnegative definite symmetric and all of the vectors of the set $\mathcal{U}_i$ lie in the row space of A.

The algorithms to be described in this report have the common feature that in constructing sequences of mutually orthogonal (or mutually A-conjugate) vectors the new linearly independent vector $y^{(i+1)}$ (or $z^{(i+1)}$) will be constructed in such a way that it is already orthogonal (or A-conjugate) to all but one or two of the vectors in the set $\mathcal{Y}_i$ (or $\mathcal{U}_i$). This permits economy in storage and in computation time since only the most recent one or two of the vectors in the set $\mathcal{Y}_i$ (or $\mathcal{U}_i$) need to be retained in storage and only the terms involving these one or two retained vectors need to be computed in Equations (2.1) or (2.2).

PART II

The Conjugate Gradient Algorithm and Variations

Chapter 3 Solving a Consistent System, $Ax=b$, where A is Symmetric and Nonnegative Definite

Let A be an $n \times n$ symmetric nonnegative definite matrix and let b be an n-vector in the column space (range space) of A. We wish to find an n-vector x satisfying

$$Ax = b \tag{3.1}$$

If Rank $(A) < n$ the solution of Problem (3.1) is nonunique. In this case there is a unique solution vector, $\hat{x}$, in the row space of A. This vector $\hat{x}$ is the minimal length solution vector for the problem and is the solution vector which the algorithm to be described constructs.

The algorithm to be described is the conjugate gradient method due to Hestenes and Stiefel (1952). Presentations of this method appear in Faddeev and Fadeeva (1963, pp. 392-405), Beckman (1960), Reid (1971), Fox (1965, pp. 208-214), and Householder (1964, pp. 139-141). This method is usually described under the assumption that the matrix A is positive definite although, as will be seen from the discussion to follow, the theory of the method is also valid for a nonnegative definite matrix if it is assumed that b is in the column space of A.

Assume the existence of an integer $k(1 \leq k \leq n)$, matrices $V_{n \times k}$ and $D_{k \times k}$ and a k-vector $\hat{p}$ such that

$$V = [v^{(1)}, \ldots, v^{(k)}] \tag{3.2}$$

(3.3) $$D = \text{Diag}\{d_1, \ldots, d_k\} \quad d_i > 0,\ i=1, \ldots, k$$

(3.4) $$V^T A V = D$$

and

(3.5) $$\hat{x} = V\hat{p}$$

Note that Equation (3.4) implies that the vectors $v^{(i)}$ are mutually A-conjugate.

If such matrices V and D are available Problem (3.1) can be attacked as follows: Left multiply Equation (3.1) by V^T obtaining

(3.6) $$V^T A x = g$$

where g is defined by

(3.7) $$g = V^T b$$

Introduce the change of variables

(3.8) $$x = Vp$$

in Equation (3.6) obtaining

(3.9) $$V^T A V p = g$$

which due to Equation (3.4) may also be written as

$$\text{(3.10)} \qquad Dp = g$$

Thus Problem (3.1) could be solved by computing $g = V^T b$, solving the diagonal system of equations $Dp = g$ for its solution vector $\hat{p}$, then computing the solution vector $\hat{x}$ for Problem (3.1) as $\hat{x} = V\hat{p}$.

The algorithm to be described constructs the A-conjugate vectors $v^{(j)}$ one at a time and as each such vector is produced its contribution to g, p, x, and the residual vector r is determined. Thus only one of the vectors $v^{(j)}$ needs to be maintained in storage at any one time.

It will be convenient to define auxiliary vectors

$$\text{(3.11)} \qquad w^{(i)} = Av^{(i)} \qquad i=1, \ldots, k$$

successive approximations to the solution vector

$$\text{(3.12)} \qquad x^{(0)} = 0$$

$$\text{(3.13)} \qquad x^{(i)} = \sum_{j=1}^{i} v^{(j)} p_j = x^{(i-1)} + v^{(i)} p_i \quad i=1, \ldots, k$$

and corresponding residual vectors

$$\text{(3.14)} \qquad r^{(0)} = b$$

(3.15) $$r^{(i)} = b-Ax^{(i)} = b - \sum_{j=1}^{i} Av^{(j)}p_j$$

$$= b - \sum_{j=1}^{i} w^{(j)}p_j = r^{(i-1)} - w^{(i)}p_i \qquad i=1,\ldots,k$$

It happens that the residual vectors $r^{(0)}$, $r^{(1)}, \ldots$ occurring in this algorithm are mutually orthogonal. The algorithm alternates between producing a vector in the orthogonal sequence $r^{(0)}$, $r^{(1)}, \ldots$ and a vector in the A-conjugate sequence $v^{(1)}$, $v^{(2)}, \ldots$. The various relations which exist between these two sequences permits the algorithm to be remarkably concise.

(3.16) Algorithm CG The Conjugate Gradient Algorithm for Solving a Consistent System Ax=b where A is Symmetric and Nonnegative Definite [Due to Hestenes and Stiefel (1952)]

Step	Description
1	$x^{(0)}:=0,\ r^{(0)}:=b,\ v^{(1)}:=b$
2	If b=0 set i:=0 and go to Step 13
3	i:=1
4	$w^{(i)}:=Av^{(i)}$
5	$p_i:=(r^{(i-1)T}r^{(i-1)}/(v^{(i)T}w^{(i)})$
6	$x^{(i)}:=x^{(i-1)}+v^{(i)}p_i$
7	$r^{(i)}:=r^{(i-1)}-w^{(i)}p_i$
8	Theoretical termination test: If $r^{(i)}=0$ go to Step 13 Practical termination test: If $\|r^{(i)}\|$ is sufficiently small go to Step 13
9	$\beta_i:=(r^{(i)T}r^{(i)})/(r^{(i-1)T}r^{(i-1)})$

Step	Description
10	$v^{(i+1)} := r^{(i)} + v^{(i)}\beta_i$
11	$i := i+1$
12	Go to Step 4
13	$k := i$
14	Stop

Figure (3.1) is provided as an aid to understanding Algorithm CG. All vectors in the first column of Figure (3.1) lie in the same one-dimensional subspace, $\mathscr{S}_1$. For general $\ell > 1$ if the vectors b, Ab, ..., $Ab^{\ell-1}$ are linearly independent let $\mathscr{S}_\ell$ denote the ℓ-dimensional subspace spanned by these vectors. Then the first k vectors in each row of Figure (3.1) are also linearly independent and span the same subspace $\mathscr{S}_\ell$.

To verify that the algorithm CG is mathematically correct we must show that the denominator in Step 5 is positive for $i \leq k$, that Step 5 defines components p_i satisfying Equation (3.10), and that the vectors $v^{(i)}$ produced at Step 10 are mutually A-conjugate. It will also be seen that the residual vectors $r^{(i)}$ are mutually orthogonal.

Assume Algorithm CG has been executed for $i=1, \ldots, \ell-1$, that the set of vectors $\{r^{(0)}, \ldots, r^{(\ell-1)}\}$ are mutually orthogonal and the set of vectors $\{v^{(1)}, \ldots, v^{(\ell)}\}$ are mutually A-conjugate. With this assumption of A-conjugacy we include the assumption that $v^{(i)T}Av^{(i)} > 0$, $i=1, \ldots, \ell$.

We also assume that

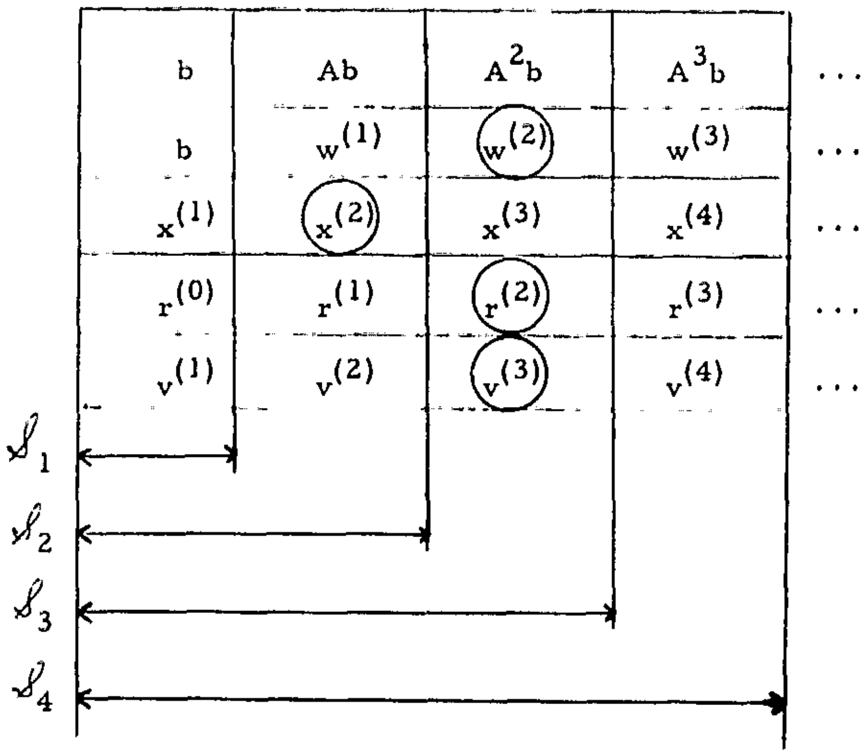

FIGURE (3.1)

Vectors and subspaces related to Algorithm CG. The four circles identify vectors which would be constructed during the second iteration of the algorithm.

$$\mathcal{S}_i = \text{Span}\{b, w^{(1)}, \ldots, w^{(i-1)}\}$$
$$= \text{Span}\{r^{(0)}, \ldots, r^{(i-1)}\}$$
$$= \text{Span}\ \{v^{(1)}, \ldots, v^{(i)}\} \qquad i=1, \ldots, \ell$$

The reader may find it convenient to refer to Figure (3. 1) and consider $\ell=2$ for definiteness. The quantities to the left of the circles result from earlier iterations and the circled quantities will be computed during the second iteration.

Consider now the quantities computed during the ℓ^{th} iteration. The denominator in Step 5 is $v^{(\ell)T}Av^{(\ell)}$ which by assumption is positive.

It must be verified that p_ℓ, computed at Step 5 satisfies Equation (3. 10), i.e. that

$$p_\ell = g_\ell/d_\ell = (v^{(\ell)T}b)/(v^{(\ell)T}Av^{(\ell)}) \tag{3.17}$$

The denominator in Equation (3. 17) is clearly identical with the denominator in Step 5. As to the numerators we have

$$v^{(\ell)T}b = v^{(\ell)T}(r^{(\ell-1)} + \sum_{j=1}^{\ell-1} w^{(j)}p_j)$$

$$= v^{(\ell)T}r^{(\ell-1)}$$

$$= (r^{(\ell-1)}+v^{(\ell-1)}\beta_{\ell-1})^T r^{(\ell-1)}$$

$$= r^{(\ell-1)T}r^{(\ell-1)}$$

Steps 6 and 7 are clearly consistent with Equations (3.13) and (3.15). If $r^{(\ell)} = 0$ the algorithm terminates, setting $k=\ell$. Otherwise with $r^{(\ell)} \neq 0$ we proceed to verify that $r^{(\ell)}$ is orthogonal to the subspace $\mathcal{S}_\ell$. Using the basis $\{r^{(0)}, \ldots, r^{(\ell-1)}\}$ of $\mathcal{S}_\ell$ it suffices to verify that $r^{(i)T} r^{(\ell)} = 0$ for $i=0, \ldots, \ell-1$.

$$r^{(i)T} r^{(\ell)} = r^{(i)T} r^{(\ell-1)} - r^{(i)T} w^{(\ell)} p_\ell \tag{3.18}$$

Both right side terms are zero for $i=0, \ldots, \ell-2$. For $i=\ell-1$ substitute the definition of p_ℓ from Step 5 into Equation (3.18) obtaining

$$\begin{aligned} r^{(\ell-1)T} r^{(\ell)} &= r^{(\ell-1)T} r^{(\ell-1)} - \frac{[r^{(\ell-1)T} w^{(\ell)}]\cdot[r^{(\ell-1)T} r^{(\ell-1)}]}{v^{(\ell)T} w^{(\ell)}} \\ &= 0 \end{aligned}$$

since

$$v^{(\ell)T} w^{(\ell)} = [r^{(\ell-1)} + v^{(\ell-1)} \beta_{\ell-1}]^T w^{(\ell)} = r^{(\ell-1)T} w^{(\ell)}$$

Next it must be shown that $v^{(\ell+1)}$ defined at Step 10 is A-conjugate to the subspace $\mathcal{S}_\ell$. We use the basis $\{v^{(1)}, \ldots, v^{(\ell)}\}$ for $\mathcal{S}_\ell$ and verify that $v^{(i)T} A v^{(\ell+1)} = 0$ for $i=1, \ldots, \ell$.

$$v^{(i)T} A v^{(\ell+1)} = v^{(i)T} A r^{(\ell)} + v^{(i)T} A v^{(\ell)} \beta_\ell \tag{3.19}$$

For i=1, ..., ℓ-1 both right side terms are zero; the first because $v^{(i)T}A$ is in $\mathcal{S}_\ell$ and thus is orthogonal to $r^{(\ell)}$ and the second because $v^{(i)}$ is in $\mathcal{S}_{\ell-1}$ and thus is A-conjugate to $v^{(\ell)}$. For i=ℓ substitute the definition of β_ℓ from Step 9 into Equation (3.19) obtaining

$$
v^{(\ell)T}Av^{(\ell+1)} = v^{(\ell)T}Ar^{(\ell)} + \frac{(v^{(\ell)T}Av^{(\ell)})(r^{(\ell)T}r^{(\ell)})}{r^{(\ell-1)T}r^{(\ell-1)}} \tag{3.20}
$$

$$
= w^{(\ell)T}r^{(\ell)} + p_\ell^{-1}r^{(\ell)T}r^{(\ell)}
$$

$$
= [w^{(\ell)} + p_\ell^{-1}r^{(\ell)}]^T r^{(\ell)}
$$

$$
= p_\ell^{-1}r^{(\ell-1)T}r^{(\ell)} = 0
$$

Finally we verify that $v^{(\ell+1)T}Av^{(\ell+1)} > 0$. Let h denote the rank of A. Since A is nonnegative definite there exists an h x n matrix R of rank h such that

$$
A = R^T R
$$

and clearly the row space of R is identical with the row (and column) space of A. Let $\mathcal{A}$ denote the row (and column) space of A. From Steps 1, 7, and 10 all of the vectors $r^{(i)}$ and $v^{(i)}$ produced by the algorithm lie in the subspace $\mathcal{A}$. Since $v^{(\ell+1)}$ is constructed at Step 10 as the sum of the nonzero vector $r^{(\ell)}$ and a vector $v^{(\ell)}\beta_i$ orthogonal to $r^{(\ell)}$ it follows that $v^{(\ell+1)}\neq 0$. Since $v^{(\ell+1)}$ is a nonzero vector in the subspace $\mathcal{Q}$ the vector $Rv^{(\ell+1)}$ must also be nonzero. It follows that $[Rv^{(\ell+1)}]^T Rv^{(\ell+1)} > 0$ or equivalently $v^{(\ell+1)T} Av^{(\ell+1)} > 0$, as was to be shown.

As is apparent from the different expressions derived for p_i and for β_i there are a number of different ways in which the CG algorithm could be implemented. There are also different trade-offs possible between storage used and counts of arithmetic operations. Reid (1971) discusses over a dozen such variations. The form in which we have stated Algorithm CG is the one preferred by Reid.

Theoretical termination of the CG algorithm occurs when $r^{(k)}=0$ with $r^{(i)}\neq 0$ for $i=0, \ldots, k-1$. Referring to Figure (3.1) we see that this means that the subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_k$ are all different but that $\mathcal{S}_{k+1} = \mathcal{S}_k$. Equivalently this means that the vector $A^k b$ lies in the subspace $\mathcal{S}_k$ spanned by $\{b, Ab, \ldots, A^{k-1}b\}$.

This implies that b is representable as a linear combination of some set of k eigenvectors, say $\{f^{(1)}, \ldots, f^{(k)}\}$, of the nonnegative definite symmetric matrix A and is not representable as a linear combination of any smaller set of eigenvectors of A. Furthermore the eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$ associated with these eigenvectors are all positive. Thus there exist nonzero numbers c_i such that

$$b = \sum_{i=1}^{k} f^{(i)} c_i$$

and the minimal length solution vector $\hat{x}$ is representable as

$$\hat{x} = \sum_{i=1}^{k} f^{(i)} (c_i/\lambda_i)$$

Note also that $\mathrm{Span}\{f^{(1)}, \ldots, f^{(k)}\} = \mathcal{S}_k$ and that $\hat{x}$ lies in $\mathcal{S}_k$ but not in $\mathcal{S}_{k-1}$.

Most commonly the value of k will be n. However k will be less than n if and only if b is orthogonal to some eigenvectors of A. In particular b will necessarily be orthogonal to some eigenvectors of A if A has any multiple eigenvalues.

Chapter 4 Solving a Least Squares Problem $Ax \cong b$

Let A be an m x n matrix and let b be an m-vector. We wish to find an n-vector x which minimizes $\|b-Ax\|$. We denote this least squares problem by the notation

$$(4.1) \qquad Ax \cong b$$

We permit either $m \geq n$ or $m < n$. If Rank (A) $< n$ the algorithm to be described constructs the unique minimal length solution vector, $\hat{x}$. This solution vector is characterized as being the only solution vector lying in the row space of A.

A vector x minimizes $\|b-Ax\|$ if and only if it satisfies the "normal equations"

$$(4.2) \qquad A^T Ax = A^T b$$

Normal equations are always consistent since the right side vector, $A^T b$, lies in the row space of A which is also the column space of the matrix $A^T A$. The ranks of $A^T A$ and A are equal and their row spaces are the same. Thus if $A^T A$ is singular the unique solution of Problem (4.2) lying in the row space of $A^T A$ is also the unique minimal length solution of Problem (4.1).

Since $A^T A$ is nonnegative definite and Problem (4.2) is consistent the conjugate gradient algorithm CG (3.16) is directly applicable to Problem (4.2). Denote the residual vector for Problem (4.2) by

$$\bar{h} = A^T b - A^T Ax = A^T (b-Ax)$$

and introduce bars on various other symbols in Algorithm CG to distinguish the application of the algorithm to Problem (4.2). Then Algorithm CG adapted to Problem (4.1) can be written briefly as

(4.3) Algorithm CGLS Conjugate Gradient Algorithm for the Least Squares Problem $Ax \cong b$

(4.4) $\bar{x}^{(0)} := 0,\ \bar{h}^{(0)} := A^T b,\ \bar{v}^{(1)} := A^T b$

If $\bar{h}^{(0)}=0$ terminate.

Do for $i:=1, 2, \ldots,$ until $\bar{h}^{(i)}=0$

(4.5) $\bar{w}^{(i)} := A^T A \bar{v}^{(i)}$

(4.6) $\bar{p}_i := \|\bar{h}^{(i-1)}\|^2 / (\bar{v}^{(i)T} \bar{w}^{(i)})$

(4.7) $\bar{x}^{(i)} := \bar{x}^{(i-1)} + \bar{v}^{(i)} \bar{p}_i$

(4.8) $\bar{h}^{(i)} := \bar{h}^{(i-1)} - \bar{w}^{(i)} \bar{p}_i$

(4.9) $\bar{\beta}_i := \|\bar{h}^{(i)}\|^2 / \|\bar{h}^{(i-1)}\|^2$

(4.10) $\bar{v}^{(i+1)} := \bar{h}^{(i)} + \bar{v}^{(i)} \bar{\beta}_i$

One may wish to have the residual vector of the least squares problem (4.1)

(4.11) $$\bar{r}^{(i)} = b - A\bar{x}^{(i)}$$

available at each iteration, possibly for use in a supplementary termination test. This may be accomplished by the following revision of the algorithm [Faddeev and Faddeeva (1963, p. 403].

(4.12) Algorithm [Alternate Form of CGLS]

$\bar{x}^{(0)} := 0$, $\bar{r}^{(0)} := b$, $\bar{h}^{(0)} := A^T b$, $\bar{v}^{(1)} := A^T b$

If $\bar{h}^{(0)} = 0$ terminate.

Do for $i := 1, 2, \ldots,$ until $\bar{h}^{(i)} = 0$

$$\bar{u}^{(i)} := A v^{(i)}$$

$$\bar{p}_i := \|\bar{h}^{(i-1)}\|^2 / \|\bar{u}^{(i)}\|^2$$

$$\bar{x}^{(i)} := \bar{x}^{(i-1)} + \bar{v}^{(i)} \bar{p}_i$$

$$\bar{r}^{(i)} := \bar{r}^{(i-1)} - \bar{u}^{(i)} \bar{p}_i$$

$$\bar{h}^{(i)} := A^T \bar{r}^{(i)}$$

$$\bar{\beta}_i := \|\bar{h}^{(i)}\|^2 / \|\bar{h}^{(i-1)}\|^2$$

$$\bar{v}^{(i+1)} := \bar{h}^{(i)} + \bar{v}^{(i)} \bar{\beta}_i$$

This latter form of the algorithm requires more storage since one must maintain the current values of the two m-vectors $\bar{u}^{(i)}$ and $\bar{r}^{(i)}$.

Chapter 5 Solving a Consistent System $Ax=b$

Let A be an $m \times n$ matrix and let b be an m-vector contained in the column space (range space) of A. Consider the problem of finding an n-vector x satisfying

$$Ax = b \tag{5.1}$$

We will permit either $m \leq n$ or $m > n$. If $\text{Rank}(A) < n$ the solution to this problem is nonunique. In this case there is a unique solution vector $\hat{x}$ lying in the row space of A. This vector $\hat{x}$ is the unique minimal length solution vector for Problem (5.1). The algorithm to be described constructs this solution vector $\hat{x}$.

Since we seek a solution vector $\hat{x}$ in the row space of A the solution vector $\hat{x}$ will be representable in the form

$$\hat{x} = A^T\hat{y} \tag{5.2}$$

for some (not necessarily unique) m-vector $\hat{y}$. Making the change of variables $x = A^T y$ in Problem (5.1) we obtain the problem

$$AA^T y = b \tag{5.3}$$

This is a consistent problem with a symmetric nonnegative definite matrix AA^T. Thus the conjugate gradient algorithm CG (3.16) can be applied to solve Problem (5.3).

The resulting algorithm for solving a consistent system $Ax=b$ may be written as follows where the notation of Algorithm CG (3.16) is changed by writing $\bar{p}$, $\bar{\beta}$, $\bar{w}$, $\bar{r}$, $\bar{q}$, $\bar{y}$, and AA^T in place of p, β, w, r, v, x, and A respectively. Furthermore, motivated by Equation (5.2) we introduce the sequence of approximate solution vectors

(5.4) $$\bar{x}^{(i)} = A^T \bar{y}^{(i)}$$

(5.5) Algorithm

$\bar{y}^{(0)} := 0,\ \bar{x}^{(0)} := 0,\ \bar{r}^{(0)} := b,\ \bar{q}^{(1)} := b$

If $\bar{r}^{(0)} = 0$ terminate.

Do for $i:=1, 2, \ldots,$ until $\bar{r}^{(i)} = 0$

$$\bar{w}^{(i)} := AA^T \bar{q}^{(i)}$$

$$\bar{p}_i := \|\bar{r}^{(i-1)}\|^2 / (\bar{q}^{(i)\,T} \bar{w}^{(i)})$$

$$\bar{y}^{(i)} = \bar{y}^{(i-1)} + \bar{q}^{(i)} \bar{p}_i$$

$$\bar{x}^{(i)} := \bar{x}^{(i-1)} + A^T \bar{q}^{(i)} \bar{p}_i$$

$$\bar{r}^{(i)} := \bar{r}^{(i-1)} - \bar{w}^{(i)} \bar{p}_i$$

$$\bar{\beta}_i := \|\bar{r}^{(i)}\|^2 / \|\bar{r}^{(i-1)}\|^2$$

$$\bar{q}^{(i+1)} := \bar{r}^{(i)} + \bar{q}^{(i)} \bar{\beta}_i$$

Eliminating the intermediate vectors $\bar{w}^{(i)}$ and $\bar{y}^{(i)}$ we obtain the algorithm in the form given by Craig (1955), p. 72, except that Craig used the opposite choice of the sign of the residual vector.

(5.6) Algorithm [Craig (1955)]

$$\bar{x}^{(0)} := 0, \ \bar{r}^{(0)} := b, \ \bar{q}^{(1)} := b$$

If $\bar{r}^{(0)} = 0$ terminate.

Do for $i:=1, 2, \ldots,$ until $\bar{r}^{(i)} = 0$

$$\bar{p}_i := \|r^{(i-1)}\|^2 / \|A^T\bar{q}^{(i)}\|^2$$

$$\bar{x}^{(i)} := \bar{x}^{(i-1)} + A^T\bar{q}^{(i)}\bar{p}_i$$

$$\bar{r}^{(i)} := b - A\bar{x}^{(i)} [\equiv \bar{r}^{(i-1)} - AA^T\bar{q}^{(i)}\bar{p}_i]$$

$$\bar{\beta}_i := \|\bar{r}^{(i)}\|^2 / \|\bar{r}^{(i-1)}\|^2$$

$$\bar{q}^{(i+1)} := \bar{r}^{(i)} + \bar{q}^{(i)}\bar{\beta}_i$$

As is noted in Faddeev and Faddeeva (1963, pp. 403-405) this algorithm can be further simplified by introducing the substitution

$$\bar{v}^{(i)} = A^T\bar{q}^{(i)}$$

Note that the vectors $\{\bar{v}^{(1)}, \ldots, \bar{v}^{(k)}\}$ are mutually orthogonal since the vectors $\{\bar{q}^{(1)}, \ldots, \bar{q}^{(k)}\}$ are mutually (AA^T)-conjugate.

We call the resulting algorithm CGC.

(5.7) Algorithm CGC Conjugate Gradient Algorithm for the Consistent System Ax=b

$$\bar{x}^{(0)} := 0, \ \bar{r}^{(0)} := b, \ \bar{v}^{(1)} := A^T b$$

If $\bar{r}^{(0)} = 0$ terminate

Do for $i:=1, 2, \ldots,$ until $r^{(i)} = 0$

$$\bar{p}_i := \|\bar{r}^{(i-1)}\|^2 / \|\bar{v}^{(i)}\|^2$$

$$\bar{x}^{(i)} := \bar{x}^{(i-1)} + \bar{v}^{(i)}\bar{p}_i$$

$$\bar{r}^{(i)} := \bar{r}^{(i-1)} - A\bar{v}^{(i)}\bar{p}_i$$

$$\beta_i := \|\bar{r}^{(i)}\|^2 / \|\bar{r}^{(i-1)}\|^2$$

$$\bar{v}^{(i-1)} := A^T\bar{r}^{(i)} + \bar{v}^{(i)}\beta_i$$

PART III
OTHER ALGORITHMS

Chapter 6 Solving a Least Squares Problem, $Ax \cong b$

Let A be an m x n matrix and let b be an m-vector. We wish to find an n-vector x which minimizes $\|b-Ax\|$. We denote this least squares problem by the notation

$$(6.1) \qquad Ax \cong b$$

Generally one would have $m \geq n$ and Rank(A) = n in such a problem. We will not make these assumptions however as they are not necessary to the mathematical development of the algorithm to be described.

If Rank (A) < n the solution to Problem (6.1) is not unique. In this case however the problem has a unique solution vector of least euclidean length. It can easily be verified that this unique minimal length solution vector lies in the row space of A and in fact is the only solution vector for Problem (6.1) lying in the row space of A. The algorithm to be described constructs the solution vector in the row space of A and thus finds the minimal length solution vector if Rank (A) < n.

Let $\hat{x}$ be the unique minimal length solution vector for Problem (6.1). Define the residual vector

$$(6.2) \qquad \hat{r} = b - A\hat{x}$$

and

$$(6.3) \qquad \hat{b} = A\hat{x} = b - \hat{r}$$

Thus b can be written as

$$(6.4) \qquad b = \hat{b} + \hat{r}$$

where $\hat{b}$ lies in the column space of A and $\hat{r}$ is orthogonal to the column space of A. The vector $\hat{b}$ is the orthogonal projection of b into the column space of A and will be referred to as the <u>projected right-side vector</u>.

Suppose there exists an integer k ($1 \leq k \leq \min\{m, n\}$) and matrices

(6.5) $$U_{m \times k} = [u^{(1)}, \ldots, u^{(k)}]$$

(6.6) $$V_{n \times k} = [v^{(1)}, \ldots, v^{(k)}]$$

(6.7) $$R_{k \times k} = \begin{bmatrix} \alpha_1 & \beta_2 & & 0 \\ & \alpha_2 & \ddots & \beta_k \\ & & \ddots & \\ 0 & & & \alpha_k \end{bmatrix} \quad (\text{all } \alpha_i > 0 \text{ and } \beta_i > 0)$$

and a k-vector, $\hat{p}$, such that

(6.8) $$U^T U = I_k$$

(6.9) $$V^T V = I_k$$

(6.10) $$AV = UR$$

(6.11) $$A^T U = VR^T$$

and

(6.12) $$\hat{x} = V\hat{p}$$

Since R is nonsingular Equations (6.9) and (6.11) imply that the column vectors of V form an orthonormal basis for a subspace, $\mathcal{V}$, of the row space of A. Similarly Equations (6.8) and (6.10) imply that the column vectors of U form an orthonormal basis for a subspace $\mathcal{U}$ of the column space (range space) of A. Equation (6.12) shows that the subspace $\mathcal{V}$ contains the solution vector $\hat{x}$. From Equations (6.3), (6.10), and (6.12) it follows that the subspace $\mathcal{U}$ contains the projected right-side vector $\hat{b}$.

Assuming the availability of the matrices U, V, and R, Problem (6.1) can be approached as follows: Introduce the change of variables

$$x = Vp \tag{6.13}$$

in Problem (6.1) and use Equation (6.10) obtaining the equivalent least squares problem

$$URp \cong b \tag{6.14}$$

Let $\bar{U}_{m \times (m-k)}$ be a matrix which when adjoined to U forms an m x m orthogonal matrix:

$$[U:\bar{U}]^T[U:\bar{U}] = I_m \tag{6.15}$$

Left multiply Equation (6.14) by $[U:\bar{U}]^T$ obtaining the equivalent least squares problem.

$$\begin{bmatrix} R \\ 0 \end{bmatrix} p \cong g \equiv \begin{array}{c} \begin{bmatrix} \tilde{g} \\ \bar{g} \end{bmatrix} \end{array} \begin{array}{l} \}\, k \\ \}\, m-k \end{array} \tag{6.16}$$

where

$$g \equiv \begin{bmatrix} \tilde{g} \\ \bar{g} \end{bmatrix} = [U{:}\overline{U}]^T b \equiv \begin{bmatrix} U^T b \\ \overline{U}^T b \end{bmatrix} \tag{6.17}$$

The least squares solution vector $\hat{p}$ for Problem (6.16) may be obtained as the solution of the upper bidiagonal nonsingular system of equations

$$Rp = \tilde{g} \tag{6.18}$$

Obtaining $\hat{p}$ from Equation (6.18) the solution vector $\hat{x}$ can then be computed from Equation (6.13).

In analogy with the conjugate gradient algorithm we wish to cast this procedure into a sequential form so that a succession of approximate solution vectors $x^{(i)}$ and associated residual vectors

$$r^{(i)} = b - Ax^{(i)} \tag{6.19}$$

can be computed as the successive vectors $u^{(i)}$ and $v^{(i)}$ are computed. This obviates the need to store old $u^{(i)}$ and $v^{(i)}$ vectors.

Equation (6.18) is not directly suitable for such a (forward) sequential procedure since the last (lower right) element of R must be determined before any components of $\hat{p}$ can be computed. However using Equations (6.13) and (6.18) we can write

$$x = Vp = VR^{-1}Rp = (VR^{-1})\tilde{g} \equiv W\tilde{g} \tag{6.20}$$

where the n x k matrix

$$W = [w^{(1)}, \ldots, w^{(k)}] \tag{6.21}$$

satisfies the linear matrix equation

$$R^T W^T = V^T \tag{6.22}$$

or

$$\begin{bmatrix} \alpha_1 & & & 0 \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ 0 & & \beta_k & \alpha_k \end{bmatrix} \cdot \begin{bmatrix} w^{(1)T} \\ \vdots \\ w^{(k)T} \end{bmatrix} = \begin{bmatrix} v^{(1)T} \\ \vdots \\ v^{(k)T} \end{bmatrix} \tag{6.23}$$

From this matrix equation one obtains the following expressions for sequential computation of the vectors $w^{(i)}$.

$$w^{(1)} = v^{(1)}/\alpha_1 \tag{6.24}$$

$$w^{(i)} = (v^{(i)} - \beta_i w^{(i-1)})/\alpha_i \qquad i=2, \ldots, k \tag{6.25}$$

Let g_i denote the i^{th} component of the m-vector g. Then g_i for $i \leq k$, is also the i^{th} component of the k-vector $\tilde{g}$ of Equation (6.18). Define the m-vectors

$$g^{(i)} = [g_1, \ldots, g_i, \underbrace{0, \ldots, 0}_{m-i}]^T \quad i=0, \ldots, m$$

and the k-vectors

$$\tilde{g}^{(i)} = [g_1, \ldots, g_i, \underbrace{0, \ldots, 0}_{k-i}]^T \quad i=0, \ldots, k$$

Motivated by Equation (6.20) define a sequence of approximate solution vectors $x^{(i)}$ by

(6.26) $$x^{(0)} = 0$$

(6.27) $$x^{(i)} = W\tilde{g}^{(i)} = \sum_{j=1}^{i} w^{(j)} g_j$$

$$= x^{(i-1)} + w^{(i)} g_i \qquad i=1, \ldots, k$$

The associated residual vectors, $r^{(i)}$, are defined as

(6.28a) $$r^{(0)} = b$$

and

(6.28b) $$r^{(i)} = b - Ax^{(i)}$$

$$= b - AVR^{-1}\tilde{g}^{(i)}$$

$$= b - URR^{-1}\tilde{g}^{(i)} \qquad \text{[Using Equation (6.10)]}$$

$$= b - U\tilde{g}^{(i)}$$

$$= b - \sum_{j=1}^{i} u^{(j)} g_i$$

$$= r^{(i-1)} - u^{(i)} g_i \qquad i=1, \ldots, k$$

We next consider formulas for computing $\tilde{g}$ and $\|A^T r^{(i)}\|$, $i=1, \ldots, k$, which depend upon a particular choice of $v^{(1)}$. The choice of $v^{(1)}$ is somewhat arbitrary as long as it is chosen to lie in the row space of A. We follow Paige (1972) in defining β_1 and $v^{(1)}$ by

(6.29) $$v^{(1)} \beta_1 = A^T b$$

where $\beta_1 = \|A^T b\|$ so that $\|v^{(1)}\| = 1$.

From Equation (6.17) we have

(6.30) $$\tilde{g} = U^T b$$

Left multiplying this equation by R^T and using Equation (6.10) and (6.29) gives

(6.31) $$R^T \tilde{g} = R^T U^T b = V^T A^T b$$

$$= \beta_1 V^T v^{(1)} = \beta_1 e^{(1)}$$

where $e^{(1)}$ denotes the first column of the k x k identity matrix.

Writing the equation $R^T \tilde{g} = \beta_1 e_1$ in terms of its components gives

$$
(6.32) \qquad \begin{bmatrix} \alpha_1 & & & 0 \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ 0 & & \beta_k & \alpha_k \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} \beta_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

from which the components g_i can be computed as

$$
(6.33) \qquad g_1 = \beta_1/\alpha_1
$$

$$
(6.34) \qquad g_i = -(\beta_i/\alpha_i)g_{i-1} \qquad i=2, \ldots, k
$$

Note that it would also be possible to compute the quantities g_i sequentially as $g_i = u^{(i)T}b$ (see Equation (6.30)) but Paige (1972) reports that Equations (6.33)-(6.34) were found to give better numerical accuracy in test cases.

In a least squares problem one does not generally expect the final residual vector, $\hat{r} = b - A\hat{x}$, to be zero. The residual vector at the solution is characterized however by the property that it is orthogonal to all of the column vectors of A. Thus the vector $h = A^T(b-Ax)$ is zero if and only if x is a solution vector for the least squares problem $Ax \cong b$. It is also true that h is the negative gradient vector with respect to x of the function $\frac{1}{2}\|b-Ax\|^2$. Define

$$
\begin{aligned}
(6.35) \qquad h^{(i)} &= A^T r^{(i)} = A^T(b - Ax^{(i)}) \\
&= A^T(b - U\tilde{g}^{(i)}) \\
&= v^{(1)}\beta_1 - VR^T\tilde{g}^{(i)}
\end{aligned}
$$

$$= V(e^{(1)}\beta_1 - R^T\tilde{g}^{(i)})$$

$$= -V e^{(i+1)}\beta_{i+1} g_i$$

$$= -v^{(i+1)}\beta_{i+1} g_i \qquad i=1, \ldots, k-1$$

while for i=0 and i=k we have $h^{(0)}=A^T b=v^{(1)}\beta_1$ and $h^{(k)}=0$. It is of interest to note that the vectors $h^{(0)}, \ldots, h^{(k-1)}$ are mutually orthogonal.

The quantities $\gamma_i=\|h^{(i)}\|$ which may be useful in monitoring the progress of the algorithm are thus expressible as

$$(6.36) \qquad \gamma_0 = \beta_1$$

$$(6.37) \qquad \gamma_i = |\beta_{i+1} g_i| \quad i=1, \ldots, k-1$$

$$(6.38) \qquad \gamma_k = 0$$

In practice k is generally not known in advance and will in fact be defined as the first value of i for which $\beta_{i+1} = 0$.

We turn now to the determination of the quantities $u^{(i)}$ and α_i for i=1, ..., k and $v^{(i)}$ and β_i for i=2, ..., k. From Equation (6.10) we obtain the equations

$$(6.39) \qquad u^{(1)}\alpha_1 = Av^{(1)}$$

$$(6.40) \qquad u^{(i)}\alpha_i = Av^{(i)} - u^{(i-1)}\beta_i \quad i=2, \ldots, k$$

and from Equation (6.11) the equations

$$(6.41) \qquad v^{(i)}\beta_i = A^T u^{(i-1)} - v^{(i-1)}\alpha_{i-1} \quad i=2, \ldots, k$$

$$(6.42) \qquad 0 = A^T u^{(k)} - v^{(k)}\alpha_k$$

If β_1 and $v^{(1)}$ are defined by Equation (6.29) and the scalar quantities β_i and α_i are determined so that the vectors $v^{(i)}$ and $u^{(i)}$ respectively have unit euclidean length (as is required by Equations (6.8) and (6.9)) then Equations (6.39)-(6.41) determine all of the remaining vectors $v^{(i)}$, $i=2,\ldots,k$, and $u^{(i)}$, $i=1,\ldots,k$.

Collecting these various formulas together leads to the following algorithm.

(6.43) Algorithm ITLS for iterative solution of the least squares problem $Ax \cong b$. [Due to C. C. Paige (1972) pp. 21-22.]

Step No.	Description
1	$x^{(0)}:=0$
2	$g_0:=-1$
3	$i:=1$
4	$\tilde{v}^{(i)}:=\begin{cases} A^Tb & \text{if } i=1 \\ A^Tu^{(i-1)}-v^{(i-1)}\alpha_{i-1} & \text{if } i>1 \end{cases}$
5	$\beta_i:=\lVert\tilde{v}^{(i)}\rVert$
6	$\gamma_{i-1}:=\lvert\beta_i g_{i-1}\rvert$
7	Theoretical termination test: If $\beta_i=0$ go to Step 17. Practical termination test: If either β_i or γ_{i-1} is sufficiently small go to Step 17.
8	$v^{(i)}:=\tilde{v}^{(i)}/\beta_i$
9	$\tilde{u}^{(i)}:=\begin{cases} Av^{(1)} & \text{if } i=1 \\ Av^{(i)}-u^{(i-1)}\beta_i & \text{if } i>1 \end{cases}$
10	$\alpha_i:=\lVert\tilde{u}^{(i)}\rVert$
11	$u^{(i)}:=\tilde{u}^{(i)}/\alpha_i$
12	$w^{(i)}:=\begin{cases} v^{(1)}/\alpha_1 & \text{if } i=1 \\ (v^{(i)}-w^{(i-1)}\beta_i)/\alpha_i & \text{if } i>1 \end{cases}$
13	$g_i:=-(\beta_i/\alpha_i)g_{i-1}$
14	$x^{(i)}:=x^{(i-1)}+w^{(i)}g_i$
15	$i:=i+1$
16	Go to Step 4
17	$k:=i-1$
18	Stop

It must be verified that the vectors $v^{(i)}$ and $u^{(i)}$ produced by this algorithm have the orthogonality properties specified by Equations (6.8) and (6.9) and that all of the numbers α_i defined by this algorithm are positive.

Assume that $\ell-1$ iterations of Algorithm (6.43) have been executed producing positive numbers $\beta^{(i)}$ and $\alpha^{(i)}$, $i=1, \ldots, \ell-1$, a set of mutually orthogonal unit n-vectors $\{v^{(1)}, \ldots, v^{(\ell-1)}\}$, and a set of mutually orthogonal unit m-vectors $\{u^{(1)}, \ldots, u^{(\ell-1)}\}$. It is obvious from Steps 4 and 8 that all of the vectors $v^{(i)}$ lie in the row space of A and from Steps 9 and 11 that all of the vectors $u^{(i)}$ lie in the column space (range space) of A.

Consider the quantities computed during the ℓ^{th} iteration.

If β_ℓ, computed at Step 5 is zero then the (theoretical) algorithm terminates, setting $k=\ell-1$. In this case we have $\beta_{k+1}=0$ and $\gamma_k=0$ which means (see Equation (6.35)) that the most recently computed approximate solution $x^{(k)}$ was in fact the unique minimum length solution $\hat{x}$ for the least squares problem, $Ax \cong b$.

If β_ℓ, computed at Step 5 is not zero, i.e., $\beta_\ell > 0$, then we must verify that the vector $\tilde{v}^{(\ell)}$ previously computed at Step 4 is orthogonal to $v^{(i)}$, $i=1, \ldots, \ell-1$. Using the formula of Step 4 with $i > 1$ the inner products to be investigated are

$$
\begin{aligned}
(6.44) \qquad v^{(i)T}\tilde{v}^{(\ell)} &= v^{(i)T}A^Tu^{(\ell-1)}-v^{(i)T}v^{(\ell-1)}\alpha_{\ell-1} \\
&= [Av^{(i)}]^Tu^{(\ell-1)}-v^{(i)T}v^{(\ell-1)}\alpha_{\ell-1} \\
&= [u^{(i)}\alpha_i+u^{(i-1)}\beta_i]^Tu^{(\ell-1)}-v^{(i)T}v^{(\ell-1)}\alpha_{\ell-1} \\
&= u^{(i)T}u^{(\ell-1)}\alpha_i+u^{(i-1)T}u^{(\ell-1)}\beta_i-v^{(i)T}v^{(\ell-1)}\alpha_{\ell-1}
\end{aligned}
$$

For $i < \ell-1$ each of the three right side terms in Equation (6.44) is zero because of the assumed mutual orthogonality of $\{u^{(1)}, \ldots, u^{(\ell-1)}\}$ and the assumed mutual orthogonality of $\{v^{(1)}, \ldots, v^{(\ell-1)}\}$.

For $i=\ell-1$ Equation (6.44) becomes

$$v^{(\ell-1)T}\tilde{v}^{(\ell)} = \alpha_{\ell-1} + 0 - \alpha_{\ell-i} = 0$$

which completes the verification that $\tilde{v}^{(\ell)}$ is orthogonal to $v^{(i)}$, $i=2,\ldots,\ell-1$. The verification that $v^{(1)T}v^{(\ell-1)}=0$ is equally straightforward.

Similarly it can be shown that $\tilde{u}^{(\ell)}$ computed at Step 9 satisfies $u^{(i)T}\tilde{u}^{(\ell)} = 0$ for $i=1,\ldots,\ell-1$. We further assert that $\tilde{u}^{(\ell)}$ is not zero and thus that $\alpha_\ell > 0$. Assume the contrary. Then

$$\begin{aligned}
0 = \tilde{u}^{(\ell)} &= Av^{(\ell)} - u^{(\ell-1)}\beta_\ell \\
&= Av^{(\ell)} - [Av^{(\ell-1)} - u^{(\ell-2)}\beta_{\ell-1}]\beta_\ell/\alpha_{\ell-1} \\
&= \cdots = \sum_{i=1}^{\ell} c_i A v^{(i)} \\
&= A \sum_{i=1}^{\ell} c_i v^{(i)}
\end{aligned} \tag{6.45}$$

where the coefficients c_i, $i=1,\ldots,\ell$ are nonzero. Since the vectors $v^{(i)}$, $i=1,\ldots,\ell$ constitute an orthonormal basis for a subspace of the row space of A the vector $z = \sum_{i=1}^{\ell} c_i v^{(i)}$ must be a nonzero vector lying in the row space of A. Such a vector must satisfy $Az \neq 0$ contradicting Equation (6.45). We conclude that $\tilde{u}^{(\ell)} \neq 0$ and $\alpha_\ell > 0$.

This completes the verification of the theoretical algorithm (6.43). In practice there remains the problem of fully specifying a satisfactory termination test at Step 7.

Some estimate, say ϵ_i, of the norm of the round-off error vector associated with the computed vector $\tilde{v}^{(i)}$ could be computed. Then the number β_i could be regarded as being sufficiently small for termination if $\beta_i \le \epsilon_i$.

Since γ_{i-1} represents an evaluation of $A^T(b-Ax^{(i-1)})$ one might define

$$\omega_i = \|A\|(\|b\| + \|A\|\cdot\|x^{(i)}\|) \tag{6.46}$$

and terminate the algorithm when $\gamma_{i-1} \le \eta\omega_{i-1}$ where η denotes the relative machine precision. (Define η to be the smallest number such that the computed value of $1+\eta$ is distinguishable from 1.)

More complex algorithmic logic might be needed. Thus if $\beta_i \le \epsilon_i$ but $\gamma_{i-1} >> \eta\omega_{i-1}$ then rather than accepting $x^{(i-1)}$ as the solution it might be useful to restart the algorithm, attacking the modified problem

$$A dx \cong b - Ax^{(i-1)} \tag{6.47}$$

to obtain a correction vector dx to be added to $x^{(i-1)}$.

We intend to study the problem of termination tests for this algorithm and treat the subject in a subsequent report.

Chapter 7 The Theoretical Equivalence of Algorithms CGLS and ITLS

We will show that the sequence of approximate solution vectors $\overline{x}^{(i)}$ generated by Algorithm CGLS (4.3) is identical (theoretically) with the sequence of approximate solution vectors $x^{(i)}$ generated by Algorithm ITLS (6.43).

It will be convenient to state the relationships represented by Algorithm CGLS in matrix form. In terms of the quantities defined in Algorithm CGLS define the matrices

(7.1) $$\overline{V} = [\overline{v}^{(1)}, \ldots, \overline{v}^{(k)}]$$

(7.2) $$\overline{H} = [\overline{h}^{(0)}, \ldots, \overline{h}^{(k-1)}]$$

(7.3) $$D = \mathrm{Diag}\{\|A\overline{v}^{(1)}\|, \ldots, \|A\overline{v}^{(k)}\|\}$$
$$\equiv \mathrm{Diag}\{d_1, \ldots, d_k\}$$

and

(7.4) $$F = \mathrm{Diag}\{\|\overline{h}^{(0)}\|, \ldots, \|\overline{h}^{(k-1)}\|\}$$
$$\equiv \mathrm{Diag}\{f_0, \ldots, f_{k-1}\}$$

Then from the orthogonality of the vectors $\overline{h}^{(i)}$ we have

(7.5) $$\overline{H}^T\overline{H} = F^2$$

and from the (A^TA)-conjugacy of the vectors $\overline{v}^{(i)}$ we have

(7.6) $$\overline{V}^TA^TA\overline{V} = D^2$$

Following Householder ((1964) pp. 2 and 139-141) define

(7.7)
$$J = \begin{bmatrix} 0 & & & 0 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}$$

Then Equations (4.8) and (4.10) can be written respectively as

(7.8)
$$\overline{H}(I-J) = A^T A \overline{V} F^2 D^{-2}$$

and

(7.9)
$$\overline{V} F^{-2}(I-J^T) F^2 = \overline{H}$$

In terms of these quantities we now define quantities, distinguished by a caret, which will be shown to satisfy the relationships of Algorithm ITLS.

(7.10)
$$\hat{V} = \overline{H} F^{-1}$$

(7.11)
$$\hat{U} = A \overline{V} D^{-1}$$

(7.12)
$$\hat{R} = D F^{-2}(I-J^T) F$$

(7.13)
$$\hat{W} = \overline{V} D^{-1}$$

Note that $\hat{R}$ is upper bidiagonal.

Using the definitions (7.10) - (7.12) and the Equations (7.5) - (7.9) it can be directly verified that the equations

$$\hat{V}^T\hat{V} = I \tag{7.14}$$

$$\hat{U}^T\hat{U} = I \tag{7.15}$$

$$A\hat{V} = \hat{U}\hat{R} \tag{7.16}$$

and

$$A^T\hat{U} = \hat{V}\hat{R}^T \tag{7.17}$$

are satisfied.

From Equation (7.10) the first column vector of the matrix $\hat{V}$ is equal to $\bar{h}^{(0)}/\|\bar{h}^{(0)}\|$ or equivalently $A^T b/\|A^T b\|$. This condition along with Equations (7.16) - (7.17) assure that the matrices $\hat{U}$, $\hat{V}$, and $\hat{R}$ are identical with the matrices U, V, and R which would be generated by Algorithm ITLS in solving the least squares problem $Ax \cong b$.

Furthermore by use of Equation (7.9) it can be directly verified that the matrix $\hat{W}$ of Equation (7.13) satisfies

$$\hat{W}\hat{R} = \hat{V} \tag{7.18}$$

In accordance with Equation (6.30) define

$$\hat{g} = \hat{U}^T b \tag{7.19}$$

Let $\hat{w}^{(i)}$ denote the i^{th} column vector of $\hat{W}$ and let $\hat{g}_i$ denote the i^{th} component of $\hat{g}$. We wish to show that the correction $\bar{v}^{(i)}\bar{p}_i$ which is added to the approximate solution vector at Equation (4.7) of Algorithm CGLS is identical with the correction vector $\hat{w}^{(i)}\hat{g}_i$ which is added to the approximate solution vector at Step 14 of Algorithm ITLS. Thus we wish to show that

(7.20) $$\bar{v}^{(i)}\bar{p}_i = \hat{w}^{(i)}\hat{g}_i \qquad i=1,\ldots,k$$

From Equation (7.13) we have $\hat{w}^{(i)} = \bar{v}^{(i)}/d_i$ and from Equations (4.6). (7.3), and (7.4) we have $\bar{p}_i = f_{i-1}^2/d_i^2$. Thus Equation (7.20) may be established by proving that

(7.30) $$f_{i-1}^2 = d_i\hat{g}_i \qquad i=1,\ldots,k$$

Introduce the k-vector $e = [1,\ldots,1]^T$ so that Equation (7.30) can be written as

(7.31) $$\begin{aligned} F^2e &= D\hat{g} & & \\ &= D\hat{U}^Tb & & \text{[Using Equation (7.19)]} \\ &= \bar{V}^TA^Tb & & \text{[Using Equation (7.11)]} \\ &= \bar{V}^T\bar{h}^{(0)} & & \text{[Using Equation (4.4)]} \end{aligned}$$

Left multiply Equation (7.8) by $\bar{V}^T$ and use Equation (7.6) obtaining

(7.32) $$\bar{V}^T\bar{H}(I-J) = F^2$$

Substitute this expression for F^2 into Equation (7.31) obtaining

$$\overline{V}^T\overline{H}(I-J)e = \overline{V}^T\overline{h}^{(0)} \tag{7.33}$$

as the equation to be verified. This equation is clearly true since $(I-J)e = [1, 0, \ldots, 0]^T$.

This completes the verification that the algorithms CGLS and ITLS produce the same sequence of approximate solutions. It follows that the vector $\overline{h}^{(i)} = A^T[b-A\overline{x}^{(i)}]$ of Equation (4.8) is identical with the vector $h^{(i)} = A^T[b-Ax^{(i)}]$ of Equation (6.35).

Chapter 8 Solving a Consistent System Ax=b

Let A be an m x n matrix and let b be an m-vector contained in the column space (range space) of A. Consider the problem of finding an n-vector x satisfying

(8.1)
$$Ax = b$$

The most common case of a consistent system of linear equations would be the case in which the matrix A is square and nonsingular. Also of practical interest is the case of a full rank underdetermined problem, i.e., $m < n$ and $\text{Rank}(A) = m$. The algorithm to be described permits either $m \le n$ or $m > n$ and does not require any restriction on the rank of A.

If $\text{Rank}(A) < n$ the solution to Problem (8.1) is nonunique. In this case there is a unique solution vector $\hat{x}$ of minimum euclidean length. This minimum length solution vector is characterized by being the only solution vector for Problem (8.1) lying in the row space of A. The algorithm to be described constructs this solution vector, $\hat{x}$.

Assume there exists an integer k ($1 \le k \le \min\{m, n\}$) and matrices

(8.2)
$$U_{m \times k} = [u^{(1)}, \ldots, u^{(k)}]$$

(8.3)
$$V_{n \times k} = [v^{(1)}, \ldots, v^{(k)}]$$

(8.4)
$$L_{k \times k} = \begin{bmatrix} \alpha_1 & & & 0 \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ 0 & & \beta_k & \alpha_k \end{bmatrix} \quad (\text{all } \alpha_i > 0 \text{ and } \beta_i > 0)$$

and a k-vector, $\hat{p}$, such that

$$U^T U = I_k \tag{8.5}$$

$$V^T V = I_k \tag{8.6}$$

$$AV = UL \tag{8.7}$$

$$A^T U = VL^T \tag{8.8}$$

and

$$\hat{x} = V\hat{p} \tag{8.9}$$

Since L is nonsingular Equations (8.6) and (8.8) imply that the column vectors of V form an orthonormal basis for a subspace $\mathcal{V}$ of the row-space of A. Similarly Equations (8.5) and (8.7) imply that the column vectors of U form an orthonormal basis for a subspace $\mathcal{U}$ of the column space (range space) of A. Equation (8.9) shows that the subspace $\mathcal{V}$ contains the solution vector $\hat{x}$. From Equations (8.1), (8.7), and (8.9) it follows that the subspace $\mathcal{U}$ contains the right-side vector b.

Assuming the availability of the matrices U, V, and L, Problem (8.1) can be approached as follows:

Introduce the change of variables

$$x = Vp \tag{8.10}$$

in Problem (8.1) and use Equation (8.7) obtaining the equivalent problem

(8.11) $$ULp = b$$

Left multiplying this equation by U^T and using Equation (8.5) gives the k x k nonsingular lower bidiagonal system

(8.12) $$Lp = g$$

where g is the k-vector defined by

(8.13) $$g = U^T b$$

A computational algorithm can thus be based on the computation of g using Equation (8.13), solving for $\hat{p}$ in Equation (8.12), and finally computing $\hat{x}$ using Equation (8.9). These steps are all directly amendable to being organized in a sequential form which uses the vectors $u^{(i)}$ and $v^{(i)}$ as they are produced.

Assuming the nontrivial case of $b \neq 0$ we follow Paige (1972) in defining $u^{(1)}$ by the equations

(8.14) $$\beta_1 = \|b\|$$

(8.15) $$u^{(1)} = b/\beta_1$$

Thus $u^{(1)}$ is in the column space (range space) of A since it was assumed that this was true of b.

The other vectors $u^{(i)}$ and $v^{(i)}$ and the elements α_i and β_i of the matrix L are sequentially determined by the unit length requirements of Equations (8.5) and (8.6) and the following equations which follow directly from Equations (8.7) and (8.8).

(8.16) $$u^{(i)}\beta_i = Av^{(i-1)} - u^{(i-1)}\alpha_{i-1} \qquad i=2,\ldots,k$$

(8.17) $$0 = Av^{(k)} - u^{(k)}\alpha_k$$

(8.18) $$v^{(1)}\alpha_1 = A^T u^{(1)}$$

(8.19) $$v^{(i)}\alpha_i = A^T u^{(i)} - v^{(i-1)}\beta_i \qquad i=2,\ldots,k$$

Comparing Equations (8.16) and (8.17) we note that the integer k is generally not known a priori but is determined as the first value of i for which $Av^{(i)} - u^{(i)}\alpha_i = 0$.

With $u^{(1)}$ chosen in the column space of A it follows from Equations (8.16), (8.18), and (8.19) that all $u^{(i)}$ will be in the column space of A and all $v^{(i)}$ will be in the row space of A. It will subsequently be verified that vectors $u^{(i)}$ and $v^{(i)}$ produced in this way necessarily satisfy the orthogonality conditions of Equations (8.5) and (8.6).

With $u^{(1)}$ defined by Equation (8.15) the vector g defined by Equation (8.13) is representable as

(8.20) $$g = \beta_1 e^{(1)}$$

where $e^{(1)}$ denotes the first column vector of the k x k identity matrix. Using this expression for g Equation (8.12) becomes

(8.21) $$Lp = \beta_1 e^{(1)}$$

which permits the components, p_i, of the solution vector $\hat{p}$ to be expressed as

(8.21) $$p_1 = \beta_1/\alpha_1$$

(8.22) $$p_i = -(\beta_i/\alpha_i)p_{i-1} \qquad i=2,\ldots,k$$

Define the sequence of k-vectors

(8.23) $$p^{(i)} = [p_1,\ldots,p_i,\underbrace{0,\ldots,0}_{k-i}]^T \qquad i=0,\ldots,k$$

Define the sequence of approximate solution vectors

(8.24) $$x^{(0)} = 0$$

(8.25) $$x^{(i)} = Vp^{(i)} = \sum_{j=1}^{i} v^{(i)}p_i = x^{(i-1)} + v^{(i)}p_i \qquad i=1,\ldots,k$$

Associated residual vectors are defined by

(8.26) $$r^{(i)} = b - Ax^{(i)} \qquad i=0,\ldots,k$$

and may be expressed as

(8.27) $$r^{(0)} = b = u^{(1)}\beta_1$$

(8.28) $$r^{(i)} = b-AVp^{(i)} = b-ULp^{(i)}$$

$$= b - U \begin{bmatrix} \beta_1 \\ 0 \\ \vdots \\ 0 \\ \beta_{i+1} p_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \leftarrow \text{row } i{+}1 \\ \\ \\ \\ \end{matrix}$$

$$= b - u^{(1)}\beta_1 - u^{(i+1)}\beta_{i+1}p_i$$

$$= -u^{(i+1)}\beta_{i+1}p_i \qquad i=1,\ldots,k-1$$

(8.29) $$r^{(k)} = 0$$

If we introduce the additional definitions, $p_0 = -1$, and $\beta_{k+1}=0$ the norm of the residual vector can be expressed as

(8.30) $$\rho_i = \|r^{(i)}\| = |\beta_{i+1}p_i| \qquad i=0,\ldots,k$$

These considerations may be organized into a computational algorithm as follows:

(8.31) Algorithm ITC For Iterative Solution of Consistent Linear Equations [Given by Paige (1972) pp. 21-22]

Step Number	Description
1	$x^{(0)}:=0$
2	$p_0:=-1$
3	$i:=1$
4	$\tilde{u}^{(i)}:=\begin{cases} b \text{ if } i=1 \\ Av^{(i-1)}-u^{(i-1)}\alpha_{i-1} \text{ if } i>1 \end{cases}$
5	$\beta_i:=\Vert\tilde{u}^{(i)}\Vert$
6	$\rho_{i-1}:=\vert\beta_i p_{i-1}\vert$
7	Theoretical termination test: If $\beta_i=0$ go to Step 16. Practical termination test: If either β_i or ρ_{i-1} is sufficiently small go to Step 16.
8	$u^{(i)}:=\tilde{u}^{(i)}/\beta_i$
9	$\tilde{v}^{(i)}:=\begin{cases} A^T u^{(1)} \text{ if } i=1 \\ A^T u^{(i)}-v^{(i-1)}\beta_i \text{ if } i>1 \end{cases}$
10	$\alpha_i:=\Vert\tilde{v}^{(i)}\Vert$
11	$v^{(i)}:=\tilde{v}^{(i)}/\alpha_i$
12	$p_i:=-(\beta_i/\alpha_i)p_{i-1}$
13	$x^{(i)}:=x^{(i-1)}+v^{(i)}p_i$
14	$i:=i+1$
15	Go to Step 4
16	$k:=i-1$
17	Stop

It must be verified that all of the vectors $u^{(i)}$ and $v^{(i)}$ produced by this algorithm have the orthogonality properties specified by Equations (8.5) and (8.6) and that all of the numbers α_i defined by this algorithm are positive.

Assume these conditions are satisfied for $i=1, \ldots, \ell-1$. Consider the quantities computed during the ℓ^{th} iteration.

If β_ℓ computed at Step 5 is zero the algorithm terminates setting $k = \ell-1$. Thus $\beta_{k+1}=0$ and $\rho_k=0$ which (see Equation (8.30)) implies that the current approximate solution vector $x^{(k)}$ is actually the unique minimal length solution vector $\hat{x}$.

If $\beta_\ell \neq 0$ then $\beta_\ell > 0$ and the orthogonality of $\tilde{u}^{(\ell)}$ relative to $u^{(1)}, \ldots, u^{(\ell-1)}$ must be verified

(8.32)
$$u^{(i)T}\tilde{u}^{(\ell)} = u^{(i)T}Av^{(\ell-1)} - u^{(i)T}u^{(\ell-1)}\alpha_{\ell-1}$$

$$= [v^{(i)}\alpha_i + v^{(i-1)}\beta_i]^T v^{(\ell-1)} - u^{(i)T}u^{(\ell-1)}\alpha_{\ell-1}$$

$$= v^{(i)T}v^{(\ell-1)}\alpha_i - u^{(i)T}u^{(\ell-1)}\alpha_{\ell-1} \qquad i=2, \ldots, \ell-1$$

For $i < \ell-1$ each term of this final expression vanishes while for $i = \ell-1$ the final expression reduces to $\alpha_{\ell-1}-\alpha_{\ell-1}=0$. The verification that $u^{(1)T}\tilde{u}^{(\ell)}=0$ is similarly straightforward.

Similarly it can be verified that the vector $\tilde{v}^{(\ell)}$ defined at Step 9 satisfies $v^{(i)T}\tilde{v}^{(\ell)}=0$ for $i=1, \ldots, \ell-1$. It remains to be shown that α_ℓ defined at Step 10 is positive.

Suppose $\alpha_\ell=0$. Then

(8.33) $$0 = \tilde{v}^{(\ell)} = A^T u^{(\ell)} - v^{(\ell-1)}\beta_\ell$$

$$= A^T u^{(\ell)} - [A^T u^{(\ell-1)} - v^{(\ell-2)}\beta_{\ell-1}]\beta_\ell/\alpha_{\ell-1}$$

$$= \cdots = \sum_{i=1}^{\ell} c_i A^T u^{(i)} = A^T \sum_{i=1}^{\ell} c_i u^{(i)}$$

Here the coefficients c_i are all nonzero and the vectors $u^{(1)}, \ldots, u^{(\ell)}$ constitute an orthonormal basis for a subspace of the column space (range space) of A. Thus the vector $z = \sum_{i=1}^{\ell} c_i u^{(i)}$ is a nonzero vector in the column space of A. It follows that $A^T z \neq 0$ which contradicts Equation (8.33). We conclude that $\alpha_\ell > 0$.

The practical termination test at Step 7 might be implemented as a comparison of β_i with some computed estimation of the norm of the round-off error vector associated with the computed vector $\tilde{u}^{(i)}$ or a comparison of ρ_{i-1} with $\eta(\|b\|+\|A\|\cdot\|x^{(i}$ where η denotes the relative machine precision.

Chapter 9 The Theoretical Equivalence of Algorithms CGC and ITC

Using the notation of Algorithm CGC (5.7) define

(9.1) $$v^{(i)} = (-1)^{i-1}\overline{v}^{(i)}/\|\overline{v}^{(i)}\| \qquad i=1,\ldots,k$$

(9.2) $$u^{(i)} = (-1)^{i-1}\overline{r}^{(i-1)}/\|\overline{r}^{(i-1)}\| \qquad i=1,\ldots,k$$

(9.3) $$\alpha_i = \|\overline{v}^{(i)}\|/\|\overline{r}^{(i-1)}\| = (\overline{p}_i)^{-1/2} \qquad i=1,\ldots,k$$

(9.4) $$\beta_1 = \|b\| = \|\overline{r}^{(0)}\|$$

(9.5) $$\beta_i = \|\overline{v}^{(i-1)}\| \cdot \|\overline{r}^{(i-1)}\|/\|\overline{r}^{(i-2)}\|^2$$

$$= (\overline{\beta}_{i-1}/\overline{p}_{i-1})^{1/2} \qquad i=2,\ldots,k$$

(9.6) $$p_0 = -1$$

(9.7) $$p_i = (-1)^{i-1}\|\overline{r}^{(i-1)}\|^2/\|\overline{v}^{(i)}\|$$

$$= (-1)^{i-1}\overline{p}_i\|\overline{v}^{(i)}\| \qquad i=1,\ldots,k$$

(9.8) $$x^{(i)} = \overline{x}^{(i)} \qquad i=0,\ldots,k$$

Since the sets of vectors $\{\overline{v}^{(1)},\ldots,\overline{v}^{(k)}\}$ and $\{\overline{r}^{(0)},\ldots,\overline{r}^{(k-1)}\}$ are each mutually orthogonal it follows that the sets $\{v^{(1)},\ldots,v^{(k)}\}$ and $\{u^{(1)},\ldots,u^{(k)}\}$ defined by Equations (9.1) and (9.2) are each mutually orthonormal. Using the equations of Algorithm CGC (5.7) it can be directly verified that the quantities defined by Equations (9.1) - (9.8) satisfy the relations of Algorithm ITC (8.31). Thus the algorithms CGC and ITC theoretically produce the same sequence of approximate solution vectors.

Since the difference between the two algorithms only involves different scale factors it is to be expected that, apart from questions of exponent overflow or underflow, the two algorithms will exhibit essentially the same numerical behaviour also.

The theoretical equivalence of these two algorithms was pointed out by Paige (1972), p. 13.

Chapter 10 Solving a Consistent System Ax=b where A is Symmetric

Let A be an n x n symmetric matrix and let b be an n-vector contained in the column space (range space) of A. We wish to find a vector x satisfying

$$Ax = b \tag{10.1}$$

In practical problems of this type the matrix A would usually be of rank n. The algorithm to be described does not require that Rank (A) = n. If Rank (A) < n the solution of Problem (10.1) is nonunique. In this case the algorithm finds the (unique) solution vector $\hat{x}$ lying in the row space of A. This is the minimal length solution vector for Problem (10.1).

Assume the existence of matrices $V_{n \times k}$ and $C_{k \times k}$ and a k-vector $\hat{p}$ such that

$$V = [v^{(1)}, \ldots, v^{(k)}] \tag{10.2}$$

$$C = \begin{bmatrix} \alpha_1 & \beta_2 & & 0 \\ & & \ddots & \beta_k \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ 0 & & \beta_k & \alpha_k \end{bmatrix}$$

$$V^T V = I_k \tag{10.4}$$

$$AV = VC \tag{10.5}$$

and

$$\hat{x} = V\hat{p} \tag{10.6}$$

If the matrices V and C are available Problem (10.1) can be attacked as follows. Make the change of variables

$$x = Vp \tag{10.7}$$

in Equation (10.1), then use Equation (10.5) obtaining

$$VCp = b \tag{10.8}$$

Left multiply Equation (10.8) by V^T using Equation (10.4).

$$Cp = V^T b \equiv g \tag{10.9}$$

Thus Problem (10.1) could be solved by first computing $g = V^T b$, next solving $Cp = g$ for $\hat{p}$, and finally computing $\hat{x} = V\hat{p}$.

Since C is a symmetric tridiagonal matrix the entire matrix C must be determined before any components of $\hat{p}$ can be computed. Thus the equation $Cp = g$ is not directly suitable for use in an algorithm which discards old vectors $v^{(i)}$ as new ones are computed.

Let Q be a k x k orthogonal matrix which on post-multiplication times C produces a k x k lower tridiagonal matrix L.

$$
(10.10) \qquad CQ = L = \begin{bmatrix} \ell_{11} & & & & 0 \\ \ell_{21} & \ell_{22} & & & \\ \ell_{31} & \ell_{32} & \ell_{33} & & \\ & \ddots & \ddots & \ddots & \\ 0 & \ell_{k,k-2} & \ell_{k,k-1} & \ell_{kk} \end{bmatrix}
$$

Define

$$
(10.11) \qquad W = VQ
$$

and

$$
(10.12) \qquad z = Q^T p
$$

Then

$$
(10.13) \qquad Lz = CQQ^T p = Cp = g
$$

and

$$
(10.14) \qquad x = Vp = VQQ^T p = Wz
$$

Thus Problem (10.1) can be solved by the following sequence of operations assuming that b, V, and C are known a priori

(10.15) $$g = V^T b$$

(10.16) $$L = CQ$$

(10.17) $$\text{Solve } Lz = g \text{ for } \hat{z}$$

(10.18) $$W = VQ$$

(10.19) $$\hat{x} = W\hat{z}$$

Each individual step of this sequence is amenable to being implemented in a sequential manner so that in fact V and C need not be known a priori but rather the column vectors of V and the elements α_i and β_i of C can be computed, used, and discarded sequentially.

This method of solving Problem (10.1) is due to C. C. Paige and M. A. Saunders (Personal correspondence, 1972).

We follow M. Saunders in defining β_1 and $v^{(1)}$ by

(10.20) $$\beta_1 = \|b\|$$

and

(10.21) $$v^{(1)} = b/\beta_1 \qquad [\text{assuming } \beta_1 \neq 0]$$

From Equation (10.5) we may write the equations

(10.22) $$v^{(2)}\beta_2 = Av^{(1)} - v^{(1)}\alpha_1$$

(10.23) $$v^{(i+1)}\beta_{i+1} = Av^{(i)} - v^{(i)}\alpha_i - v^{(i-1)}\beta_i \qquad i=2, \ldots, k-1$$

and

(10.24) $$0 = Av^{(k)} - v^{(k)}\alpha_k - v^{(k-1)}\beta_k$$

The numbers β_{i+1} may be computed as normalization factors to assure that the vectors $v^{(i+1)}$ have unit length as required by Equation (10.4). The numbers α_i can be computed as

(10.25) $$\alpha_i = v^{(i)T}Av^{(i)}$$

which is the condition (see Equation (2.1)) which assures that $v^{(i+1)}$ computed by Equation (10.22) or (10.23) will be orthogonal to $v^{(i)}$.

The integer k will generally not be known in advance and may (theoretically) be determined as the first value of i for which the right side of Equation (10.23) is zero. With k so determined it can be verified that the vectors $v^{(1)}, \ldots, v^{(k)}$ produced by use of Equations (10.21) - (10.23) form an orthonormal basis for a subspace of the column space of A. This verification makes essential use of the symmetry of A.

It is pointed out by Paige and Saunders that the computation of the orthogonal set of vectors $v^{(i)}, i=1, \ldots, k$, using Equations (10.22)-(10.23) is a method due to Lanczos (1950 and 1952).

The orthogonal matrix Q will be (implicitly) constructed as the product

$$Q = \tilde{G}_1 \tilde{G}_2 \cdots \tilde{G}_{k-1} \tag{10.26}$$

where

$$\tilde{G}_i = \begin{array}{c} \left[\begin{array}{ccc} I & 0 & 0 \\ 0 & G_i & 0 \\ 0 & 0 & I \end{array}\right] \begin{array}{l} i-1 \\ 2 \\ k-i-1 \end{array} \\ \begin{array}{ccc} \underbrace{\quad}_{i-1} & \underbrace{\quad}_{2} & \underbrace{\quad}_{k-i-1} \end{array} \end{array} \qquad i=1, \ldots, k-1 \tag{10.27}$$

$$G_i = \begin{bmatrix} c_i & s_i \\ s_i & -c_i \end{bmatrix} \qquad i=1, \ldots, k-1 \tag{10.28}$$

and

$$c_i^2 + s_i^2 = 1 \tag{10.29}$$

Each matrix G_i effects a nontrivial transformation on a particular 3 x 2 or 2 x 2 submatrix of the appropriate intermediate matrix which arises in the process of transforming the symmetric tridiagonal matrix C to the lower tridiagonal matrix L. This action may be expressed as follows:

$$\bar{\ell}_{11} = \alpha_1 \tag{10.30}$$

(10.31) $$\bar{\ell}_{21} = \beta_2$$

(10.32) $$\begin{bmatrix} \bar{\ell}_{i,i} & \beta_{i+1} \\ \bar{\ell}_{i+1,i} & \alpha_{i+1} \\ 0 & \beta_{i+2} \end{bmatrix} \cdot G_i = \begin{bmatrix} \ell_{i,i} & 0 \\ \ell_{i+1,i} & \bar{\ell}_{i+1,i+1} \\ \ell_{i+2,i} & \bar{\ell}_{i+2,i+1} \end{bmatrix} \qquad i=1,\ldots,k-2$$

(10.33) $$\begin{bmatrix} \bar{\ell}_{k-1,k-1} & \beta_k \\ \bar{\ell}_{k,k-1} & \alpha_k \end{bmatrix} \cdot G_{k-1} = \begin{bmatrix} \ell_{k-1,k-1} & 0 \\ \ell_{k,k-1} & \ell_{k,k} \end{bmatrix}$$

As each additional row of L is determined by these equations an additional component z_i of the vector $\hat{z}$ satisfying Equation (10.17) can be determined. Note that

(10.34) $$g = \begin{bmatrix} \beta_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

due to Equations (10.4), (10.15), and (10.21). Thus the equations for the components z_i are

(10.35) $$z_1 = \beta_1 / \ell_{11}$$

(10.36) $$z_2 = -\ell_{21} z_1 / \ell_{22}$$

and

(10.37) $$z_i = -(\ell_{i,i-2} z_{i-2} + \ell_{i,i-1} z_{i-1}) / \ell_{ii} \qquad i=3, \ldots, k$$

Define the vectors

(10.38) $$z^{(i)} = [z_1, \ldots, z_i, \underbrace{0, \ldots, 0}_{k-i}] \qquad i=0, 1, \ldots, k$$

and, motivated by Equation (10.19), define a sequence of approximate solution vectors

$$x^{(i)} = Wz^{(i)} \qquad i=0, 1, \ldots, k$$

The associated residual vectors are expressible as

(10.39) $$\begin{aligned} r^{(i)} &= b - Ax^{(i)} = b - AWz^{(i)} \\ &= b - AVQz^{(i)} = b - VCQz^{(i)} \\ &= b - VLz^{(i)} = v^{(1)}\beta_1 - VLz^{(i)} \\ &= V[e^{(1)}\beta_1 - Lz^{(i)}] \qquad i=0, 1, \ldots, k \end{aligned}$$

from which we may write

(10.40) $$r^{(0)} = b = v^{(1)}\beta_1$$

(10.41) $$r^{(i)} = -[v^{(i+1)} \vdots v^{(i+2)}] \cdot \begin{bmatrix} \ell_{i+1,i-1} & \ell_{i+1,i} \\ 0 & \ell_{i+2,i} \end{bmatrix} \cdot \begin{bmatrix} z_{i-1} \\ z_i \end{bmatrix} \quad i=1,\ldots,k-1$$

(10.42) $$r^{(k)} = 0$$

Interpretation of Equation (10.41) for i=1 requires the definitions $\ell_{2,0}=0$ and $z_0=0$ while for i=k-1 one must define $v^{(k+1)}=0$ and $\ell_{k+1,k-1}=0$.

The norms of the residual vectors are expressible as

(10.43) $$\rho_0 \equiv \|r^{(0)}\| = \beta_1$$

(10.44) $$\rho_i \equiv \|r^{(i)}\| = [(\ell_{i+1,i-1}z_{i-1}+\ell_{i+1,i}z_i)^2+(\ell_{i+2,i}z_i)^2]^{1/2}$$

for i=1, . . . , k-1

(10.45) $$\rho_k \equiv \|r^{(k)}\| = 0$$

where again we define $\ell_{2,0}=0$, $z_0=0$, and $\ell_{k+1,k-1}=0$.

Combining these equations appropriately one can obtain the following algorithm

(10.46) Algorithm ICSE Iterative Solution of a Consistent Symmetric System of Linear Equations [Originated by C. C. Paige and M. A. Saunders, personal communication, 1972]

Step	Description
1	$x^{(0)} := 0$
2	$\beta_1 := \|b\|$
3	If $\beta_1 = 0$ set $k := 0$ and go to Step 24
4	$v^{(1)} := b/\beta_1$
5	$u^{(1)} := v^{(1)}$
6	QUIT:=FALSE
7	$i := 1$
8	$y^{(i)} := Av^{(i)}$
9	$\alpha_i := v^{(i)T} y^{(i)}$
10	$\tilde{v}^{(i+1)} := \begin{cases} y^{(1)} - \alpha_1 v^{(1)} & \text{if } i=1 \\ y^{(i)} - \alpha_i v^{(i)} - \beta_i v^{(i-1)} & \text{if } i > 1 \end{cases}$
11	$\beta_{i+1} := \|\tilde{v}^{(i+1)}\|$
12	Theoretical termination test: If $\beta_{i+1} = 0$ set QUIT:=TRUE Practical termination test: If β_{i+1} is sufficiently small set QUIT:=TRUE
13	$\begin{cases} \begin{bmatrix} \bar{\ell}_{11} \\ \bar{\ell}_{21} \end{bmatrix} := \begin{bmatrix} \alpha_1 \\ \beta_2 \end{bmatrix} & \text{if } i=1 \\ \begin{bmatrix} \ell_{i,i-1} & \bar{\ell}_{i,i} \\ \ell_{i+1,i-1} & \bar{\ell}_{i+1,i} \end{bmatrix} := \begin{bmatrix} \bar{\ell}_{i,i-1} & \alpha_i \\ 0 & \beta_{i+1} \end{bmatrix} G_{i-1} & \text{if } i > 1 \end{cases}$

Step	Description
14	If $i=1$ go to Step 16; If $i=2$ set $\rho_1 := [(\ell_{21}z_1)^2+(\ell_{31}z_1)^2]^{1/2}$; If $i>2$ set $\rho_{i-1} := [(\ell_{i,i-2}z_{i-2}+\ell_{i,i-1}z_{i-1})^2+(\ell_{i+1,i-1}z_{i-1})^2]^{1/2}$
15	Practical termination test: If ρ_{i-1} is sufficiently small set $k:=i-1$ and go to Step 24.
16	$\ell_{ii} := (\bar{\ell}_{ii}^2 + \beta_{i+1}^2)^{1/2}$
17	$z_i := \begin{cases} \beta_1/\ell_{11} \text{ if } i=1 \\ -\ell_{21}z_1/\ell_{22} \text{ if } i=2 \\ -(\ell_{i,i-2}z_{i-2}+\ell_{i,i-1}z_{i-1})/\ell_{ii} \text{ if } i>2 \end{cases}$
18	If QUIT=<u>FALSE</u> set $v^{(i+1)} := \tilde{v}^{(i+1)}/\beta_{i+1}$, $c_i := \bar{\ell}_{ii}/\ell_{ii}$, $s_i := \beta_{i+1}/\ell_{ii}$, $G_i := \begin{bmatrix} c_i & s_i \\ s_i & -c_i \end{bmatrix}$, and $[w^{(i)}, u^{(i+1)}] := [u^{(i)}, v^{(i+1)}]G_i$
19	If QUIT=<u>TRUE</u> set $w^{(i)} := u^{(i)}$
20	$x^{(i)} := x^{(i-1)} + w^{(i)}z_i$
21	If QUIT=<u>TRUE</u> set $k:=i$ and go to Step 24
22	$i:=i+1$
23	Go to Step 8
24	Here the algorithm is finished with the solution vector $x^{(k)}$.

References

F. S. Beckman (1960), The Solution of Linear Equations by the Conjugate Gradient Method, pp. 62-72 of Mathematical Methods for Digital Computers, ed. by Ralston and Wilf, Wiley.

E. J. Craig (1955), The N-Step Iteration Procedures, Journal of Math. and Physics, 34, (1955), 64-73.

D. K. Faddeev and V. N. Faddeeva (1963), Computational Methods of Linear Algebra, Freeman.

L. Fox (1965), An Introduction to Numerical Linear Algebra, Oxford Univ. Press.

G. Golub and W. Kahan (1965), Calculating the Singular Values and Pseudoinverse of a Matrix, J. SIAM Numer. Anal., Ser B, 2, 205-224.

M. R. Hestenes and E. Stiefel (1952), Methods of Conjugate Gradients for Solving Linear Systems, J. Res. Nat. Bur. Standards, 49, 409-436.

A. S. Householder (1964), The Theory of Matrices in Numerical Analysis, Blaisdell

C. Lanczos (1950), An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, J. Res. Nat. Bur. Standards, 45, 255-282.

C. Lanczos (1952), Solution of Systems of Linear Equations by Minimized Iterations, J. Res. Nat. Bur. Standards, 49, 33-53.

C. C. Paige (1972), Bidiagonalization of Matrices and Solution of Linear Equations, Stanford Univ. Report No. CS-295, 28 pp.

J. K. Reid (1971), On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations, pp. 231-254 in Large Sparse Sets of Linear Equations, ed. by J. K. Reid, Academic Press.

NASA — JPL — Coml., L.A., Calif.